



저작자표시-비영리-동일조건변경허락 2.0 대한민국

이용자는 아래의 조건을 따르는 경우에 한하여 자유롭게

- 이 저작물을 복제, 배포, 전송, 전시, 공연 및 방송할 수 있습니다.
- 이차적 저작물을 작성할 수 있습니다.

다음과 같은 조건을 따라야 합니다:



저작자표시. 귀하는 원저작자를 표시하여야 합니다.



비영리. 귀하는 이 저작물을 영리 목적으로 이용할 수 없습니다.



동일조건변경허락. 귀하가 이 저작물을 개작, 변형 또는 가공했을 경우에는, 이 저작물과 동일한 이용허락조건하에서만 배포할 수 있습니다.

- 귀하는, 이 저작물의 재이용이나 배포의 경우, 이 저작물에 적용된 이용허락조건을 명확하게 나타내어야 합니다.
- 저작권자로부터 별도의 허가를 받으면 이러한 조건들은 적용되지 않습니다.

저작권법에 따른 이용자의 권리는 위의 내용에 의하여 영향을 받지 않습니다.

이것은 [이용허락규약\(Legal Code\)](#)을 이해하기 쉽게 요약한 것입니다.

[Disclaimer](#)

공학박사 학위논문

User Profiling for Personalized
Search & Partnership Match

개인화 검색 및 파트너십 선정을
위한 사용자 프로파일링

2014 년 2 월

서울대학교 대학원

치의과학과 의료경영과정보학 전공

Harshit Kumar

User Profiling for Personalized Search & Partnership Match

개인화 검색 및 파트너십 선정을 위한 사용자
프로파일링

지도교수 김 홍 기

이 논문을 공학박사 학위논문으로 제출함
2013 년 12 월

서울대학교 대학원
치의과학과 의료경영과정정보학 전공
Harshit Kumar

Harshit Kumar의 박사 학위논문을 인준함
2014 년 2 월

위 원 장 김 형 주 (인)

부위원장 김 홍 기 (인)

위 원 이 상 구 (인)

위 원 김 명 기 (인)

위 원 임 동 혁 (인)

User Profiling for Personalized Search & Partnership Match

Adviser Hong-Gee Kim

Submitting a doctoral thesis of Computer
Science and Engineering
December 2013

Graduate School of Seoul National University
Department of Dental Science
Harshit Kumar

Confirming the doctoral thesis written by
Harshit Kumar
February 2014

Chair Hyung-Joo Kim (인)

Vice Chair Hon-Gee Kim (인)

Examiner Sang-Goo Lee (인)

Examiner Myeng-Ki Kim (인)

Examiner Dong-Hyuk Im (인)

Abstract

User Profiling for Personalized Search & Partnership Match

The secret of change is to focus all of your energy not on fighting the old, but on building the new. - Socrates

The automatic identification of user intention is an important but highly challenging research problem whose solution can greatly benefit information systems. In this thesis, I look at the problem of identifying sources of user interests, extracting latent semantics from it, and modelling it as a user profile. I present algorithms that automatically infer user interests and extract hidden semantics from it, specifically aimed at improving personalized search. I also present a methodology to model user profile as a buyer profile or a seller profile, where the attributes of the profile are populated from a controlled vocabulary. The buyer profiles and seller profiles are used in partnership match.

In the domain of personalized search, first, a novel method to construct a profile of user interests is proposed which is based on mining anchor text. Second, two methods are proposed to build a user profile that gather terms from a folksonomy system where matrix factorization technique is explored to discover hidden relationship between them. The objective of the methods is to discover latent relationship between terms such that contextually, semantically, and syntactically related terms could be grouped together, thus disambiguating the context of term usage. The profile of user interests is also analysed to judge its clustering tendency and clustering accuracy. Extensive evaluation indicates that a profile of user interests, that can correctly or precisely disambiguate the context of user query, has a significant impact on the personalized search quality. In the domain of partnership match, an ontology termed as partnership ontology is proposed. The attributes or concepts, in the partnership ontology, are features representing context of work. It is used by users to lay down their requirements as buyer profiles or seller profiles. A semantic similarity measure is defined to compute a ranked list of matching seller profiles for a given buyer profile.

Keywords : User Modelling, User Interests, User Preferences, Personalized Search, Partnership Match.

Student ID: 2010-31376

Contents

List of Figures	vii
List of Tables	xii
1 Introduction	1
1.1 User Profiling for Personalized Search	9
1.1.1 Motivation	10
1.1.2 Research Problems	11
1.2 User Profiling for Partnership Match	18
1.2.1 Motivation	19
1.2.2 Research Problems	24
1.3 Contributions	25
1.4 System Architecture - Personalized Search	29

1.5	System Architecture - Partnership Match	31
1.6	Organization of this Dissertation	32
2	Background	35
2.1	Introduction to Social Web	35
2.2	Matrix Decomposition Methods	40
2.3	User Interest Profile For Personalized Web Search - Non Folksonomy based	43
2.4	User Interest Profile for Personalized Web Search - Folksonomy based	45
2.5	Personalized Search	47
2.6	Partnership Match	52
3	Mining anchor text for building User Interest Pro- file: A non-folksonomy based personalized search	56
3.1	Exclusively Yours'	59
3.1.1	Infer User Interests	61
3.1.2	Weight Computation	64
3.1.3	Query Expansion	67
3.2	Exclusively Yours' Algorithm	68
3.3	Experiments	71

3.3.1	DataSet	72
3.3.2	Evaluation Metrics	73
3.3.3	User Profile Efficacy	74
3.3.4	Personalized vs. Non-Personalized Results .	76
3.4	Conclusions	80
4 Matrix factorization for building Clustered User Interest Profile: A folksonomy based personalized search 82		
4.1	Aggregating tags from user search history	86
4.2	Latent Semantics in UIP	90
4.2.1	Computing the tag-tag Similarity matrix . .	90
4.2.2	Tag Clustering to generate <i>svdCUIP</i> and <i>modSvdCUIP</i>	98
4.3	Personalized Search	101
4.4	Experimental Evaluation	103
4.4.1	Data Set and Experiment Methodology . . .	103
4.4.1.1	Custom Data Set and Evaluation Metrics	103
4.4.1.2	AOL Query Data Set and Evaluation Metrics	107

4.4.1.3	Experiment set up to estimate the value of k and d	107
4.4.1.4	Experiment set up to compare the proposed approaches with other approaches	109
4.4.2	Experiment Results	111
4.4.2.1	Clustering Tendency	111
4.4.2.2	Determining the value for dimension parameter, k , for the Custom Data Set	113
4.4.2.3	Determining the value of distinctness parameter, d , for the Custom data set	115
4.4.2.4	CUIP visualization	117
4.4.2.5	Determining the value of the dimension reduction parameter k for the AOL data set	119
4.4.2.6	Determining the value of distinctness parameter, d , for the AOL data set	120

4.4.2.7	Time to generate <i>svdCUIP</i> and <i>modSvd-</i> <i>CUIP</i>	122
4.4.2.8	Comparison of the <i>svdCUIP</i> , <i>modSvd-</i> <i>CUIP</i> , and <i>tfIdfCUIP</i> for different classes of queries	123
4.4.2.9	Comparing all five methods - Im- provement	124
4.4.3	Discussion	126
5	User Profiling for Partnership Match	133
5.1	Supplier Selection	137
5.2	Criteria for Partnership Establishment	140
5.3	Partnership Ontology	143
5.4	Case Study	147
5.4.1	Buyer Profile and Seller Profile	153
5.4.2	Semantic Similarity Measure	155
5.5	Discussion	160
5.6	Conclusions	162
6	Conclusion	164
6.1	Future Work	167

CONTENTS

6.1.1	Degree of Personalization	167
6.1.2	Filter Bubble	168
6.1.3	IPR issues in Partnership Match	169
	Bibliography	170
	Appendices	193
.1	Pairs of Query and target URL	194
.2	Examples of Expanded Queries	197
.3	An example of svdCUIP, modSvdCUIP, tfIdfCUIP	198

List of Figures

1.1	User Profiling features for various classes of Web Systems	2
1.2	User Profiling for Personalized Search and Partnership Match	9
1.3	Percentage of partnerships that are successful, partial successful, and failures	20
1.4	Reasons that cause failure of partnership	21
1.5	Percentage of companies who has formed joint ventures with other companies.	22
1.6	Key benefits of partnering.	23
1.7	A system architecture for building a <i>CUIP</i> and its application to personalized search	29

LIST OF FIGURES

1.8	An archetype for Partnership match, showing the flow of processes	31
3.1	System Architecture of Exclusively Your's	60
3.2	Set U represents URLs returned by a search engine and set V represents URLs clicked or downloaded by the user. On the right (b), URLs $h_1, h_2, h_3, \dots, h_n$ are hub URLs for URL V_i	61
3.3	A Snapshot of Exclusively Yours' user interface	69
3.4	(a) Display URLs, snippet and title (b) extracts anchor text and its surrounding text from hub URLs.	71
3.5	Efficacy of UIP constructed using different methods	76
3.6	(a) Cumulative Gain (CG) Curve for an individual user query (b) Discounted Cumulative Gain (DCG) for an individual user query.	77
3.7	Average Discounted Cumulative Gain (DCG) Curve and (b) Average Cumulative Gain (CG)	78
3.8	(a) Average Rank vs. each department (b) Average Rank vs. Search Engine	79
4.1	System Architecture of CUIP based Personalized Search	86

LIST OF FIGURES

4.2	Dendrogram visualization for similarity matrix <i>modSim</i>	100
4.3	Automatic Evaluation Methodology	108
4.4	Number of Clusters vs. average Silhouette Coefficient plot for <i>svdCUIP</i> and <i>modSvdCUIP</i>	112
4.5	A comparison of different value combinations of <i>k</i> and <i>d</i> Vs. average Silhouette Coefficient for <i>svdCUIP</i> average linkage	114
4.6	A comparison of different value combinations of <i>k</i> and <i>d</i> vs average Silhouette Coefficient for <i>modSvdCUIP</i> average linkage	115
4.7	A comparison of different value combinations <i>k</i> and <i>d</i> vs <i>AverageFscores</i> for the <i>modSvdCUIP</i> (when <i>k</i> =30,40) and the <i>svdCUIP</i> (when <i>k</i> =90,100) for average linkage.	117
4.8	Estimating the values of dimension parameter for <i>svdCUIP</i> and <i>modSvdCUIP</i> using the Improvement as an evaluation metric	119
4.9	Estimating the values of distinctness parameter for <i>tJdfCUIP</i> , <i>svdCUIP@90</i> , <i>modSvdCUIP@100</i> using Improvement as an evaluation metric.	120

LIST OF FIGURES

4.10	Average time to generate <i>svdCUIP</i> and <i>modSvdCUIP</i>	122
4.11	Comparing the Percentage Increase of the <i>tfIdfCUIP</i> , <i>svdCUIP</i> , <i>modSvdCUIP</i> for two classes of queries: vague and self-evident.	124
4.12	Comparing the Improvement of <i>tfIdfUIP</i> , <i>tfUIP</i> , <i>tfIdfCUIP-0.09</i> , <i>svdCUIP-90-0.13</i> , <i>modSvdCUIP-100-0.63</i>	127
5.1	Partnership Ontology: concepts and properties that define relationship between them. Various other stan- dard ontologies like Dublin Core, FOAF, Geo, VCard etc are also imported.	142
5.2	Seller Profiles for this study: Seller1 and Seller2 . .	149
5.3	Seller Profiles for this study: Seller3 and Seller4 . .	150
5.4	Seller Profiles for this study: Seller5	151
5.5	An example to demonstrate construction of user pro- file (Buyer Profile) - concepts shown here are derived from the Partnership Ontology	152

LIST OF FIGURES

5.6 A reduced version of buyer profile - truncated to fit
in here. The features that buyer does not choose
during profile construction are removed to save space. 156

5.7 Search Results showing the ranked list of matching
seller profiles to a given buyer profile. 160

List of Tables

1.1	A snapshot of an exemplary <i>UIP</i> obtained from (Noll and Meinel, 2007) work on personalized search based on folksonomy	13
1.2	leftmost column shows the original rank of search results from google in middle column. Rightmost column shows the adjustment in the rank of search results after application of <i>UIP</i>	14

2.1	A comparison summary of the proposed approaches with the other similar approaches that uses folksonomy for personalized search. (a)Source of terms for constructing a <i>UIP</i> , (b) Web document Representation, (c) Similarity Measure, (d)First-Order Co-occurrence, (e) Second-Order Co-occurrence, (f)Clustering of terms in a <i>UIP</i> , (g) <i>UIP</i> and resource length normalization factor	46
3.1	Top three Hub URLs for the IMDB URL	63
3.2	Terms extracted from the <i>Hub URL</i> ₁	63
4.1	Clicked Web documents and tags attached to the documents	87
4.2	A user context derivable from Table 4.1	88
4.3	Clusters obtained by applying HAC on similarity matrices <i>Sim</i> ₃ and <i>modSim</i> ₃ for <i>k</i> =3 and <i>d</i> =0.35	100
4.4	Example of cluster structure	118
4.5	Comparing the MRRs of <i>tfIdfUIP</i> , <i>tfUIP</i> , <i>tfIdfCUIP</i> , <i>svdCUIP</i> , and <i>modSvdCUIP</i>	126

LIST OF TABLES

5.1	List of Concepts produced by amalgamating contribution of various research work's in domain of Partnership Establishment.	144
1	List of Self-evident query and target URL pairs . . .	194
2	List of vague query and target URL pairs	195

1

Introduction

The only true wisdom is in knowing you know nothing. - Socrates

Adaptive Web Systems (AWS), belongs to the class of user-adaptive software systems (Brusilovsky, Kobsa, and Nejdl, 2007) that largely depends on the existence of *user profile*. The user profile is a representation of information about a user that is essential to an adaptive system to provide the adapted effect, i.e., to recommend meaningful and relevant products or results for different users. For instance, for a user query *apple*, a search engine may return search results related to *apple* as a fruit, *apple* as an iPhone or iPad, or *apple* as in the context of eye. According to wikipedia ¹, a user profile is a collection of personal data associated with a specific user. A user profile can be manifested in different forms in different domains. What data is included in a user profile depends on the domain or the application. It may include user's interests, user's preferences, user's goals or plans, and user's likes or dislikes. To create and maintain

¹http://en.wikipedia.org/wiki/User_profile

an up-to-date user profile, a Web system collects data from various resources that may include implicitly observing user interaction or explicitly requesting direct input from the user. This process is called as **user profiling**.

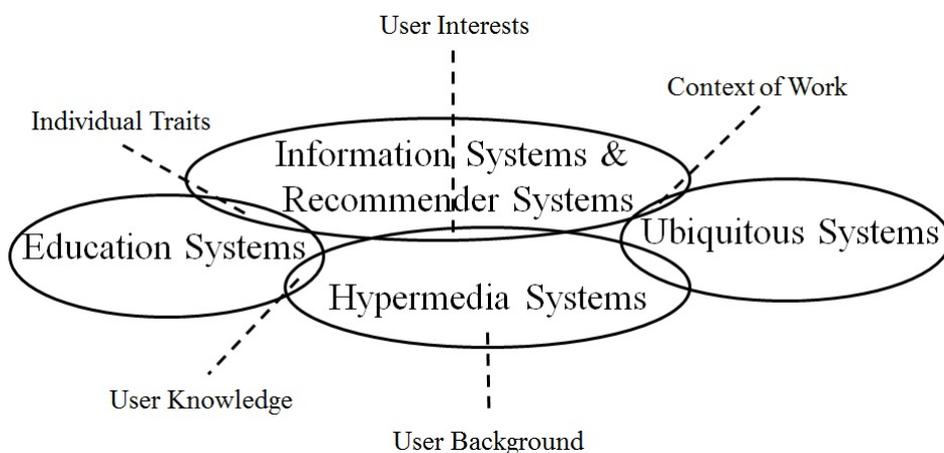


Figure 1.1: User Profiling features for various classes of Web Systems

One common feature across all Web systems is the enactment of user profiles to acculturate the system’s behaviour to individual users. User Profiles represent information about users that is essential to remodel and improve the functionality of the system with the ultimate goal of improving user experience. Web Systems have surveyed a plethora of approaches to user profiling from exploring how to accumulate user data, storing it, organizing it, and keep it up-to-date. Most of the Web systems focus on features to model information

about the users for representing a user profile. The widely used features are user knowledge, user interests, goals, background, individual traits, and context of work. Each individual Web system capitalizes on a subset of these features to model a user profile, the selection of features largely depends on the domain of interest, refer Figure 1.1. Feature based modelling of user profile aims to model user's specific features such as knowledge, interests, goals, etc. It is based on user's interaction with the system. During the user interaction, these features may change, so is the user profile. Therefore, in feature based modelling, a user profile is always up-to-date. A contrarian approach, which is an age old approach, is stereotype user profiling [163,164]. Stereotype user profiling aims to cluster all possible user types into several groups, called stereotypes. The goal of stereotype user profiling is to map individual user features to a particular group. Both methods, personalized search and partnership match, proposed in this work are based on feature based user profiling.

Since this work is focused on feature based user profiling, we will now focus on various features that are essential to building a user profile. The most widely used features are:

1. **Knowledge** : It is the most commonly used feature in Web based education systems for modelling a user profile. The user's knowledge is a variable feature, in the sense that a user's knowledge is upgrading, or deteriorating, or is staying constant. This warrants that a particular Web based education system has to recognize the changes in user's knowledge level and update the user profile accordingly. Some examples of

web based education systems that uses user profile are WITS (Okazaki, Watanabe, and Kondo, 1997), ILESA(López, Millán, Pérez-de-la Cruz, and Triguero, 1998), Web-PVT(Tsiriga and Virvou, 2003). The simplest form of user knowledge based profile is a scalar profile. It estimates the level of user domain knowledge on a scale of 0 to 5 (quantitative) or as one of the classes of good, average, fair, poor, none (qualitative). Different versions of the Web page are presented to individual users based on their levels of knowledge(Beaumont, 1994, Boyle and Encarnacion, 1998, Brailsford, Stewart, Zakaria, and Moore, 2002). The shortcoming of scalar based user profile is low precision. The user knowledge of any domain can vary for different part of the domain. It would require dividing a domain into sub-domains, and further eliciting from the user his/her knowledge of each sub-domain. These scores are then syndicated to generate a combined score for the whole domain. One of the challenge in this kind of methodology is to estimate all sub-domains for a given domain. Further, it would also require various representations of each Web page for different knowledge levels in different sub-domains. This could be a challenging task.

2. **Interests** : User Interests has had always been the most important constituent of a user profile in information systems or recommender systems that dealt with overwhelming amount of information. The personalized search methods proposed in this thesis are also based on user interests.

As a first step, the aim is to automatically identify user interests. Further, the proposed methods identify and group similar user interests into group. The similarity is identified in terms of syntactic or semantic or contextual. Early Web based education systems paid no attention to user interests. However, in the recent decade, the situation has changed dramatically. There is a competition between user interests and user knowledge when it comes to what constitutes an essential part of a user profile. This is essentially due to the increased user interactions with Web systems that are mostly interest driven, such as news systems(Abel, Gao, Houben, and Tao, 2011), electronic stores(Rossi, Schwabe, and Guimarães, 2001), museum(Rennick-Egglestone, Whitbrook, Leygue, Greensmith, Walker, Benford, Schnädelbach, Reeves, Marshall, Kirk, et al., 2011). The predominant approach to model user interests in a profile is through the weighted vector of terms or keywords, and this approach is still widely used. In contrast to keyword level approach to building user profile, another most recent approach is concept based approach to profiling user interests. Concept based approach to user profiling provides a more accurate representation compared to keyword based approach. For instance, a news personalization system can profile user interests on distinct topics, that could be based on location, genres, named entities, and so on(Abel, Gao, Houben, and Tao, 2011). In closed Web systems such as museum, even ontologies can be employed for mapping user interests to concepts in the ontologies. Whereas, in Open systems such as news personaliza-

tion, employing ontologies can be an overwhelming task. In nut-shell, open Web systems uses keyword based approach to profiling user interests and closed Web systems uses concept based approach to profiling user interests.

3. **Goals and Tasks** : User's goals and tasks represent immediate information need of a user. The user goal is most changeable user feature; it changes with each session and often changes within one session also. Planning and sequencing systems model user goals to build a user profile(Brusilovsky, 1992, McArthur, Stasz, Hotta, Peter, and Burdorf, 1988, McCalla, Bunt, and Harms, 1986, Vassileva, 1990). User's immediate information need is also diagnosed by information retrieval systems(Brajnik, Guida, and Tasso, 1987). A hierarchy of user goal is developed, and it is assumed that at one point of time user has a particular goal. This warrants identifying user goal to one of the goals in the goal hierarchy. Based on the current goal, relevant Web pages are recommended to the user or are adapted to user information needs. A popular example of goal based Web system is ADAPTS(Brusilovsky and Cooper, 2002). This system has a small hierarchy of goals. The system observes user behaviour to detect the current user goal, and depending on that, Web pages are adapted before presenting them to the user. This system was developed for aircraft maintenance operations.
4. **Background** : The user's background comprises of user's location, lan-

guage, profession, etc. For instance, clinical decision support systems can classify a user's knowledge of medical terminology to pre-defined set of categories. For each category different Decision Aids (DA) are developed. Based on the category of the user, the relevant DA is presented to the user. Another example of user adapted Web systems is the categorization of users by their language ability (native or non-native), followed by choosing the appropriate version of the content for them (Kay and Kummerfeld, 1994). Background information of a user is also used in Web based navigation support systems (Vassileva, 1996).

5. **Individual Traits** : The user's individual traits is an amalgamation of various user features that define a user as an individual. Some of the user features are personality traits (introvert/extrovert), cognitive styles (holist/serialist), cognitive factors (working memory capacity, focus), and learning styles. Similar to user background, user individual traits consist of stable features that don't change suddenly. To identify user individual traits, a psychological interview or tests are required. It has been widely acknowledged by research in IR to model user individual traits and use for personalization. Psychological literature has immense discussion with great width and depth of individual traits, however, in the field of user profiling, the interest is largely in cognitive styles and learning styles. Cognitive styles in layman terms mean an individual habit about how he/she organizes and represent information (Riding

and Rayner, 1998). Learning styles refer to how an individual learns or absorbs information. This feature is used for education based personalization systems. In the context of personalized museum guides(Krüger, Baus, Heckmann, Kruppa, and Wasinger, 2007), a user profile is used that consists of user's personality factors. Another research on adaptive Web page generation(Tarpin-Bernard and Habieb-Mammar, 2005) is based on user's lower level cognitive abilities.

6. **Context of work** : The context of work, is rather a new feature, that is being used to build a user profile in Web systems. In the beginning, it was introduced to build Web systems and later expanded into the area of personalized Web systems. In personalized clinical decision support systems, it adds a new dimension of human personal context, i.e., blood pressure, mood, cognitive load, etc. Another dimension to context in user profiling is the user platform or device, this is called as device oriented context. This kind of context is very dominant in mobile and ubiquitous computing. Finally, one more dimension that has been added to context is the context of work. This is the context of work that the user is dealing with. It is called a user oriented context. For instance, in the partnership match, we have taken the context of work as user profiling. Since the context of work is to find partners, and partners are represented as buyers and sellers, therefore, we have modelled two different types of profiles: buyer profile and seller profile.

1.1 User Profiling for Personalized Search

In this thesis, refer Figure 1.2, a user profile is manifested as User Interest Profile and buyer profile or seller profile in the domain of Personalized Search and Partnership Match, respectively. Chapter 3 and 4 demonstrates how a user profile is manifested as a User Interest Profile (*UIP*) for personalized search. Chapter 5 demonstrates how a user profile is manifested as a buyer profile or seller profile for partnership match



Figure 1.2: User Profiling for Personalized Search and Partnership Match

1.1 User Profiling for Personalized Search

A search engine returns the most relevant search results matching a user query, however, it often fails to judge the user query intent or user interests. To improve the quality of search results, the system needs to understand different aspects associated with a user query: one is user interest, and the other is query intent. A user model, built from user interactions with the Web and folksonomy, plays a bigger part in disambiguating query intent by taking clues from user interests. User interests can be considered as contextual variants that may help to disambiguate user query intent when the original query is vague or there are too many search results that a user has to wade through to find the most relevant ones. Moreover, the amount of information available on-line is

1.1 User Profiling for Personalized Search

increasing exponentially. While this information is a valuable resource, its sheer volume limits its value. Many research projects and companies are exploring the use of personalized applications that manage this deluge by tailoring the information presented to individual users. These applications need to gather, and exploit, some information about individuals in order to be effective.

1.1.1 Motivation

The most prevalent way for computer users to find the required information is to surf the Web and search through Internet pages. Having various available free search engines, such as Google, Bing, and Yahoo, makes Internet searching the first and easiest way to find relevant content. In this case, the user expresses his information need as a small set of keywords and receives a ranked list of documents. Having a list of retrieved documents, however, is not enough for the user to find the exact information that he is looking for. The user has to spend more time with these documents to extract the exact information need from the large retrieved documents. Such a manual processing step is not possible without spending a large amount of time.

Personalization has emerged as an appealing approach when dealing with the issues caused by the variation of on-line behaviors and individual differences observed in user interests, information needs, search goals, query contexts, and others (Ioannis, Konstantinos, and Joemon, 2010). Personalized Search Engines return different results for different users even though the input query is same. The results are differentiated based on the input query by the user

1.1 User Profiling for Personalized Search

and user interests. In certain scenarios, search results are re-ranked based on each user interests. These leads to improved search quality, and it needs additional efforts which indicates that developing a personalized search system needs studies beyond search engine development. This goal is mainly achieved using a combination of important techniques:

1. **Natural Language Processing** methods which analyze input documents and user search history to build user profile.
2. **Information Retrieval** methods which retrieve a set of relevant documents from the input corpus and re-rank them based on the user profile
3. **Data Mining** methods which clusters the terms in the user profile so that contextually similar concepts are grouped together thus disambiguating polysemy and synonymy. Also, it requires matrix factorization methods to discover latent information that is useful to calculate the similarity between terms in the user profile.

To achieve such a system, a pipeline of different components is required which constructs the whole architecture of the personalized search system. This dissertation mainly focusses on building such system.

1.1.2 Research Problems

To achieve a personalized system, one of the core requirement is to build a profile of user interests. Existing research works in user modeling use the phrase

1.1 User Profiling for Personalized Search

user profile which can be misleading; a *user profile*¹ often means user personal information, such as name, address, and age. Our intention is not to collect user personal information, instead, our goal is to collect user interests. We, therefore, coin a new term, User Interest Profile (*UIP*), which we believe is more appropriate because such a profile reflects user interests and not user personal information.

The primary research problem, addressed in Chapters 3 and 4, is building a User Interest Profile (*UIP*) that consists of user interests and their context. The *UIP* is further used for re-ranking search results, thus providing personalized results to a user. User interests are inferred from user search behavior which is obtained by mining user's search history or URLs clicked by the user during his/her search sessions. Given a list of clicked URLs, interesting re-search problems are: How to summarize them to generate a list of terms, How to eliminate noisy terms, and How to determine context of terms that represent user interests?

Most recent works, (David, Iván, and Joemon, 2010, Noll and Meinel, 2007, Xu, Bao, Fei, Su, and Yu, 2008), related to personalized search use folksonomy to build a *UIP* from the clicked web pages; however, there are some inherent limitations which we discuss next, and propose solutions to remedy them.

Limitation 1: The concepts, that make a *UIP*, are collected from the resource profiles of clicked URLs emanating from user search sessions. A *UIP*

¹http://en.wikipedia.org/wiki/User_profile

1.1 User Profiling for Personalized Search

is further used in other search sessions to re-rank search results by calculating cosine similarity between the resource profiles of search results URLs with all concepts in a *UIP*. To ease the exposition, consider a scenario from (Noll and Meinel, 2007) work on user profiling for personalized search. Table 1.1 shows the *UIP*, for a user, constructed using folksonomies.

Table 1.1: A snapshot of an exemplary *UIP* obtained from (Noll and Meinel, 2007) work on personalized search based on folksonomy

Concept Name	Concept-Weight
open source	13
programming	19
proprietary	2
research	10
security	21
semantic web	34

1.1 User Profiling for Personalized Search

Table 1.2: leftmost column shows the original rank of search results from google in middle column. Rightmost column shows the adjustment in the rank of search results after application of *UIP*.

Original Rank	URL	Re-ranked
1	securityfocus.com/	1 •
2	microsoft.com/security/	7 ↓
3	microsoft.com/technet/security/def/...	3 •
4	dhs.gov/	10 ↓
5	whitehouse.gov/homeland/	9 ↓
6	windowsitpro.com/WindowsSecurity/	8 ↓
7	ssa.gov/	5 ↑
8	w3.org/Security/	4 ↑
9	cert.org/	2 ↑
10	nsa.gov/	6 ↑

One can infer from the *UIP* in Table 1.1 that the user interests are *security*, *programming*, *research*, and *semantic web*. Table 1.2 shows the effect of *UIP* on the ranking of search results. The leftmost column of Table 1.2 shows the original ranking of search results returned by the Google search engine for a user query *security*. The rightmost column of Table 1.2 shows the adjustment in the ranks of the search results after personalization based on *UIP* in Table 1.1. Meticulously observing the leftmost column and the rightmost column

1.1 User Profiling for Personalized Search

of Table 1.2, one can infer that the URLs related to terms *computing* and *security* are promoted to the top. However, there exists no reasoning that explains the quantitative effect of the terms, in a *UIP*, on the ranking of search results. That is, why a particular URL gets promoted more than the other URL, when both the URLs are relevant to the same term, say term *security*. Or, why a URL is promoted more than the other URL, even though one of the URL is less related to the user query compared to the other URL. Authors mention that the URL of US Security and Administration is promoted even though it is not related to concepts *computing* and *security*. We offer the following explanation; some terms, in a *UIP*, even though not related to user query *security*, but because they are present in a *UIP*, contributes to the ranking score of URLs in the search results. The term, in this case *insurance* in the *UIP* (not shown in Table 1.1 but authors mentioned in their paper that concept *insurance* exists in the *UIP*), has a false positive effect on the ranking of URL. The reason, why the URL of US Security Administration is promoted, is because of the incapability of the system to judge the context of user query *security*. Note that, the terms, in a *UIP*, may have false positive or false negative effect on the re-ranking of URLs, which is actually uncalled for. We claim that the related terms in a *UIP* should be clustered together and work as a cluster; since *security* and *insurance* are unrelated terms, URLs that are re-ranked based on the term *security* should not be effected by the presence of term *insurance* in a *UIP*. In other words, the term *insurance* should not contribute towards the re-ranking score of the search results obtained from a

1.1 User Profiling for Personalized Search

search engine for a user query *security*. The terms, in a *UIP*, that are related to concept *security* can definitely help to disambiguate it, for ex: if a item *IT* is clustered together with a term *security*, and both are used in conjunction for computation of re-ranking score with the resource profiles of search results; the computed re-ranking score will help to positively promote the rank of URLs related to terms *IT* and *security*, and demote the rank of URLs related to terms *security* and *device*, or *security* and *administration*, or alike. In the existing work, terms in a *UIP* are not clustered into groups, therefore whether the terms are related to a user query or not, they anyway participate in the computation of re-ranking score. We propose to cluster the related terms in a *UIP* resulting in a clustered *UIP*. A cluster of terms, related to a user query instead of all terms in the *UIP*, is used for calculating the re-ranking score of URLs. This allows to consider terms in a matching cluster to a user query for re-ranking score computation with the resource profiles of search results.

The experiment results verify our claim that clustering the terms, present in a *UIP* thus generating a clustered *UIP* (*CUIP*), has many advantages; it helps to disambiguate context of a user query, mitigate polysemy problem and synonymy problem, reduces the time complexity of re-ranking, and improves the precision of the search results. The clustering of concepts in a *UIP* allows to disambiguate user interests by associating the context which is otherwise latent.

Limitation 2: A resource like URL is tagged by many users. For each URL, a resource profile is created. But since, users don't tag resources religiously;

1.1 User Profiling for Personalized Search

it may be possible that a resource profile, of a particular URL, has tags with higher tag-weights while others don't. Popular URLs, compare to less popular URLs, are tagged by many users. Hence, popular URLs have more number of tags with high value of tag weights. The existing work does not take into account the biases of tagging by users. To alleviate such biases, we propose to normalize the value of tag weights associated with tags in a resource profile. For illustration purpose, consider resource profiles of two URLs: $URL_1 = \{java : 50, programming : 10\}$, $URL_2 = \{java : 5, programming : 1\}$. The resource profile of URL_1 and URL_2 have similar tags, but tags in resource profile of URL_1 have higher value of tag-weight. Existing work is based on the hypothesis that value of tag-weight reflect the importance of tags in a *UIP*. However, if we normalize the tag-weights of tags in a *UIP*, it gives a different picture. After normalization, resource profile of URLs will be as follows: $URL_1 = \{java : 5, programming : 1\}$, $URL_2 = \{java : 5, programming : 1\}$. This suggests that both tags are equally important for URL_1 and URL_2 .

Limitation 3: We experimented with the search query log of users and observed that users exhibit sporadic search behavior. We find that two factors, viz. user search behavior and URL popularity, effect the number of tags and value of tag-weights in a *UIP*. This further means that, some users search actively while others are intermittently active. Active users' *UIP* consists of tags with high value of tag-weights while non-active users' *UIP* contains tags with low value of tag-weights. Existing works, assume that, users whose *UIPs* have

1.2 User Profiling for Partnership Match

tags with high value of tag-weights are more interested in those tags. While non active users whose *UIPs* have tags with low value of tag-weights are less interested in those tags. The biases of user search activity can lead to invalid personalized search results(Wang and Jin, 2010). We propose to annul or dilute the biases due to sporadic user search behavior by normalizing the tag-weights in each *UIP*.

1.2 User Profiling for Partnership Match

In order to maximize the advantages and minimize the negative effects of globalization and growing interdependence, it is imperative for SMEs (Small and Medium Enterprises) in developing countries to forge partnerships with big enterprises in developed regions. However, the partnership establishment process is a rough ride; it comes with its own set of hurdles. A survey by PricewaterhouseCoopers (PwC) indicates that 44% of the partnerships were unsuccessful. In this dissertation, we refer to research literature to find out various features that are involved during partnership establishment process. Based upon a review, we select features that form core concepts in a partnership establishment process. These concepts along with their related properties are modeled as an ontology, termed as Partnership Ontology. A user that could represent a big enterprise or a SME (Small and Medium Enterprise) can use the partnership ontology to lay down their requirements as a buyer profile and/or a seller profile. A semantic similarity measure is defined to compute a ranked list of matching

1.2 User Profiling for Partnership Match

seller profiles given a buyer profile. We illustrate the devised methodology of partnership establishment process by an example using a case study.

1.2.1 Motivation

Partnership is a voluntary collaborative agreement between two or more parties in which all participants agree to work together to achieve a common purpose or undertake a specific task and to share risks, responsibilities, resources, competencies and benefits. Meaningful partnerships are the foundation for success. Partnerships are what enable many companies to make continuous improvements. By sharing with others, one can direct their resources and capabilities to projects what they consider most important. The selection of the right partners is a critical element of an Extended Enterprise (EE) strategy. Although most companies understand the importance of selecting the right partner, they often do not spend enough time understanding their individual needs and defining their requirements. As a result there is a greater risk of an incorrect selection decision, which may ultimately lead to a failed partnership. This has negative ripple effect for other parties along the EE from down through the supply chain and forward through the customer chain. A survey taken by Business Consultants has revealed that 49% of the partnerships are very successful, 44% results in partial success and 7% are a failure, shown in Figure 1.3. The most common causes of failure cited by CEOs are: cultural differences, poor or unclear leadership, and poor integration process. The above are the major reasons, though there is plethora of factors that affect a partnership establishment process.

1.2 User Profiling for Partnership Match

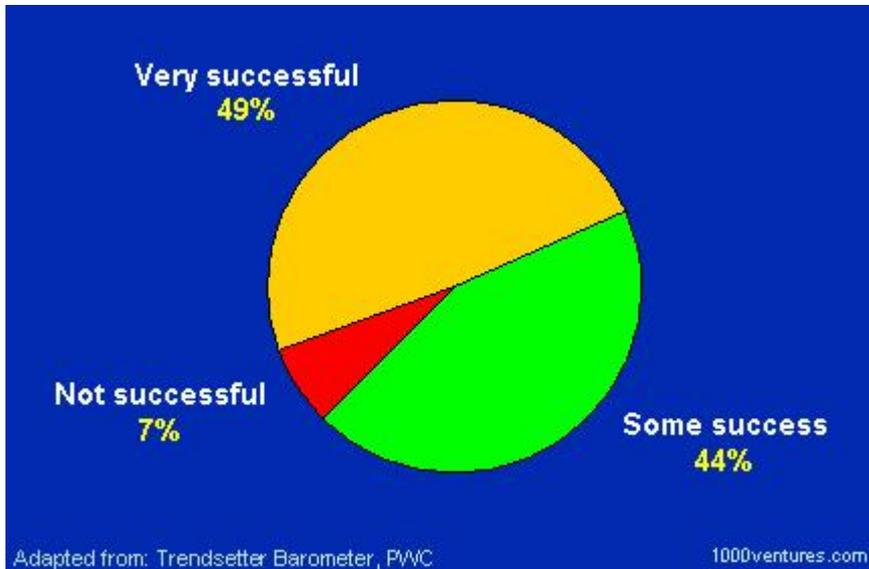


Figure 1.3: Percentage of partnerships that are successful, partial successful, and failures

Figure 1.4 below shows that 49% of the failures are due to poor or unclear leadership, another 49% are due to cultural differences, whereas 46% of the failures are due to poor integration processes. Analysis of these results gives enough reason to improve the partnership matching process so as to reduce the partial success and failure partnerships. Another survey carried out by PwC (PriceWaterHouseCoopers) interviewed CEOs of 239 Fortune 500 companies, refer Figure 1.5; results from the survey shows that 56% of the companies in US have partnered over the past 3 years. These companies have partnered with large companies (41%), large MNCs (28%), large domestic companies (22%),



Figure 1.4: Reasons that cause failure of partnership

small companies (29%), university (7%), and federal lab (3%). The interviewees cite three major benefits of partnering, based upon their own experiences: increased profit opportunities (88%), secured competitive position (87%), and increased sale of existing products (80%), refer Figure 1.6. Two other benefits are creation of more new products or lines of business, cited by 66%; and better operations or technologies (60%). The emergence of globalization process in industrial scenario is forcing users to consider forming network partnerships and collaborations, such as EE, in order to achieve a sustainable competitive advance and growth. However, the success rate of partnerships is found to be low, which is due to the selection of unsuitable partners. Therefore, part-

1.2 User Profiling for Partnership Match

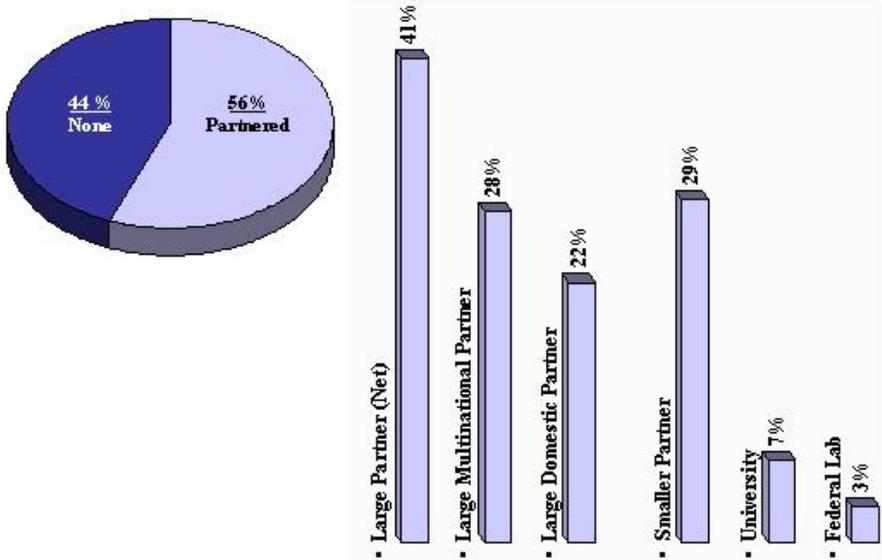
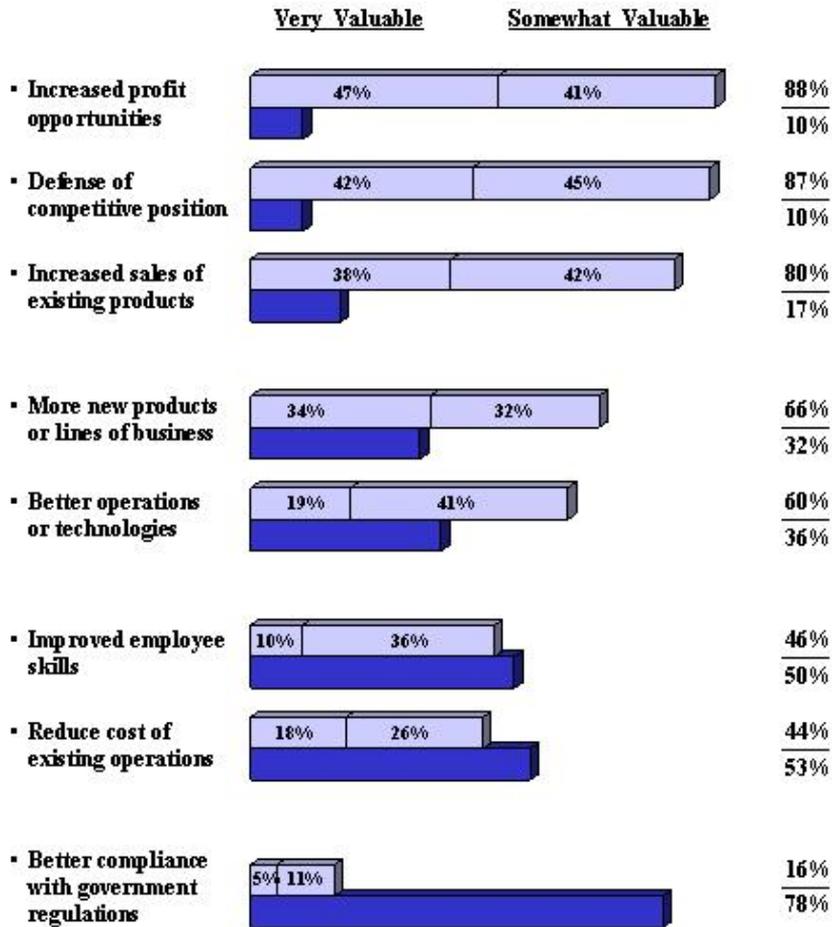


Figure 1.5: Percentage of companies who has formed joint ventures with other companies.

nership match plays a key role in the success of a partnership. A meticulous examination of the key components in the partnership match reveals that a very few formal partnership match process exist, and those that do are not sufficient to support partnership match effectively; results in Figure 1.3 vouch the said claim. This is further complicated when an ODM from a developed country, for instance South Korea, seeks a partner from a developing country, such as India. Thus a critical question is how globally separated organizations can be supported to establish an EE partnership that increases the chances of the optimum set of partners being selected, while being conducted effectively

1.2 User Profiling for Partnership Match

and efficiently.



Base: 239 Firms Involved in Partnering Over Past 3 Years

Very or Somewhat Valuable
 Not Too or Not At All Valuable

Figure 1.6: Key benefits of partnering.

1.2.2 Research Problems

The projects that operate within inter-enterprise environments additionally face the problem that different information models are likely to be used by different partners. Engineers working within a particular organization will inevitably develop their own vocabulary for particular activities and these will need to be adjusted to be more practical and to meet the requirements of different collaborating partners. Hence, when two different partners are brought together, two common types of problem can occur in communications that share and exchange information, firstly, the same term is being applied to different concepts (semantic problem), secondly, different terms may be used to denote the same entity (syntax problem). This problem is popularly known as integration problem in literature.

The objective of the proposed Partnership Match is to explore the fundamental problem: How distributed organizations be supported to establish an EE (Extended Enterprise) partnership that increases the chances of the optimal partner being selected, while being conducted efficiently and effectively without any syntax or semantic disambiguation.

The key hypothesis of partnership match is that, a process perspective is employed in order to help users representing organizations effectively manage their distributed partnership establishment process. This structured approach enables both users and associated users' profile information to be presented in a generic machine readable format, a mechanized matching process to take place

and partnership management to be managed effectively.

In order to explore such hypothesis, this thesis intends to answer several problems: how to effectively model user's profile; i.e. what should be the key components that form a user profile, how to make user profile machine readable so that it can be processed and further reasoned by the machine, and to define semantic similarity measures for compare user profiles.

By solving these problems, Partnership Match will allow the development of new services to manage social interactions, establishing a partnership process between users (buyers and suppliers), creating a conducive collaboration environment, and a structured approach to managing the generation, and machine to machine manipulation, of request and offer profiles as part of partnership match process. These services will open new business opportunities for networked enterprises to provide new products/services. Partnership Match will develop generic services, applicable across different domains, and specifically explore new business opportunities in manufacturing and engineering SMEs.

1.3 Contributions

The main contribution of this dissertation is to improve user satisfaction in the context of search results and partnership match.

To this aim, for personalized search, we propose three methods to model user profile and also propose an automatic evaluation method. And, for partnership match, we propose an ontology that can be used for building user profiles that

can be further modelled as buyer profiles or seller profiles.

1. The first method for personalized search is a non-folksonomy based method. It is called as Exclusively Yours'. It uses anchor text to build a *UIP*. We also propose how to compute term-weights for terms in the *UIP* and also how to find matching terms in a *UIP* for a given user query.
2. The second method for personalized search is a folksonomy based method. It uses Singular Value Decomposition (SVD), a matrix factorization method to discover latent information, to generate a *svdCUIP*.
3. The third method for personalized search is also folksonomy based method. It is a variation of SVD, modSVD, to generate a *modSvdCUIP*. *modSvdCUIP* represents a better cluster structure as compared to *svdCUIP*.
4. One of the impediments in the personalized search research area is evaluation. Researchers find it difficult to get access to user query logs, and even if they can get access to it, evaluation also requires users' involvement to evaluate the quality of search results. We propose an automatic evaluation method that doesn't requires user involvement at any stage. Thus our proposed methods, or for that matter, any personalized search method can be evaluated using our proposed evaluation method.
5. For partnership match, I propose an ontology to provide a machine readable representation of buyer and seller profiles. A semantic similarity

measure is also proposed that ranks seller profiles for a given buyer profile. The system is implemented as a web service that can be hosted on a web server, thus providing an easy access to users. The proposed methodology is unique in the sense that ontologies are employed and vector space model is used so as to provide a solid systematic approach which is also mathematically proven. The major innovation of the proposed methodology is that the UNSPSC ontology provides a unique code for manufacturing skills that helps in disambiguation of any product or services. Classifying products and services with a common coding scheme facilitates commerce between buyers and sellers and is becoming mandatory in the new era of electronic commerce.

The existing works, for construction of a *UIP*, assume that a user is registered with one or more social network service. We don't make such assumption. The proposed system observes and analyzes a user search behavior to construct his/her profile. Thus our system is applicable to all users with no dependency on a particular search engine or a particular social network service (SNS). The system architecture developed in this work can be used with any search engine or any SNS, provided the search engine or SNS has its open access API available.

In addition to the proposed methods for building a *CUIP*, we also propose an automatic evaluation method to test the proposed methods with the baseline search and folksonomy based personalized search approaches. In our evaluations, we found that the improvement in the ranking scores of the target URLs

1.3 Contributions

for the *modSvdCUIP* based personalized search were better than all the other methods; the *modSvdCUIP* approach showed improvement of 71.6%, 27.8%, 12%, 6.6%, and 8.1% over the baseline (Lucene Search), *tfIdfUIP*, *tfUIP*, *tfIdfCUIP*, and *svdCUIP* approaches, respectively.

1.4 System Architecture - Personalized Search

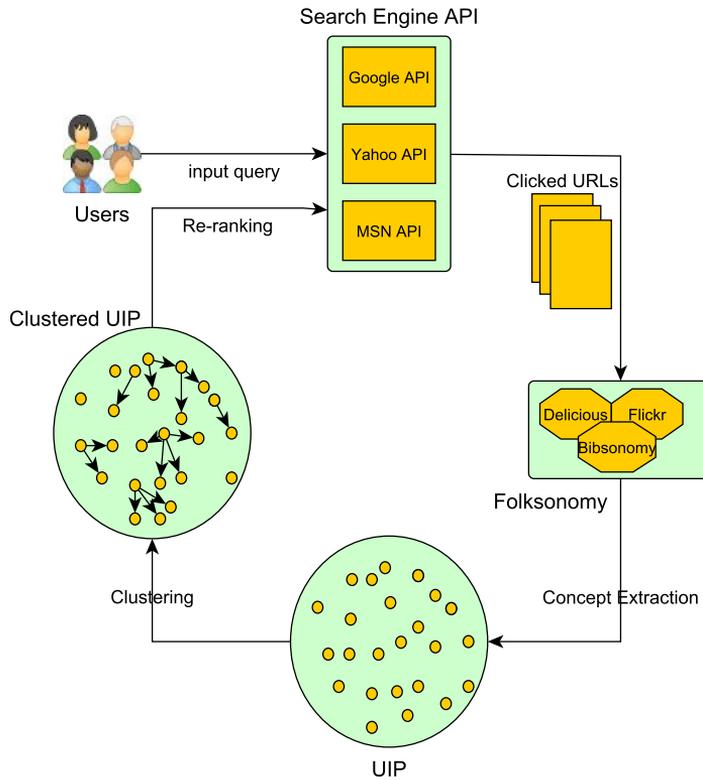


Figure 1.7: A system architecture for building a *CUIP* and its application to personalized search

This section describes the system architecture for building Clustered User Interest Profile (*CUIP*), and how the *CUIP* is used for re-ranking search results for personalization. It begins with the explanation of the sequence diagram

1.4 System Architecture - Personalized Search

that encompasses various modules of the system; collecting user search history, extracting and mining user interests from user search history to build a *UIP*, clustering concepts to build a *CUIP*, and finally using *CUIP* for personalized search. Figure 1.7 shows various modules and their connections using a sequence diagram.

A user session begins with a given input query. The input query is submitted to a search engine, and the output is a ranked list of URLs. Furthermore, based on the relevancy of the output ranked list of URLs, a user clicks on URLs of his/her interest. A list of clicked URLs, which we believe reflect user interests, is processed to extract concepts. To extract concepts for a given URL, it is submitted to a social bookmarking service which returns a list of tags and tag-weights. The list of tags and tag-weights are imported to construct a *UIP*. The extracted terms are further manipulated using factorization techniques and clustering algorithms to discover a set of meaningful concept clusters. The final clusters of terms represent a *CUIP*. Each concept in a cluster has a weight associated with it reflecting its importance in the cluster. The *CUIP* is further used for re-ranking search results to provide a personalized search result set for a given input user query in the following search sessions. Figure 1.7 shows three search engine APIs: Google API, Yahoo API, MSN API; this only means any one of the API can be used to obtain search results. Similar reasoning goes for the folksonomy.

1.5 System Architecture - Partnership Match

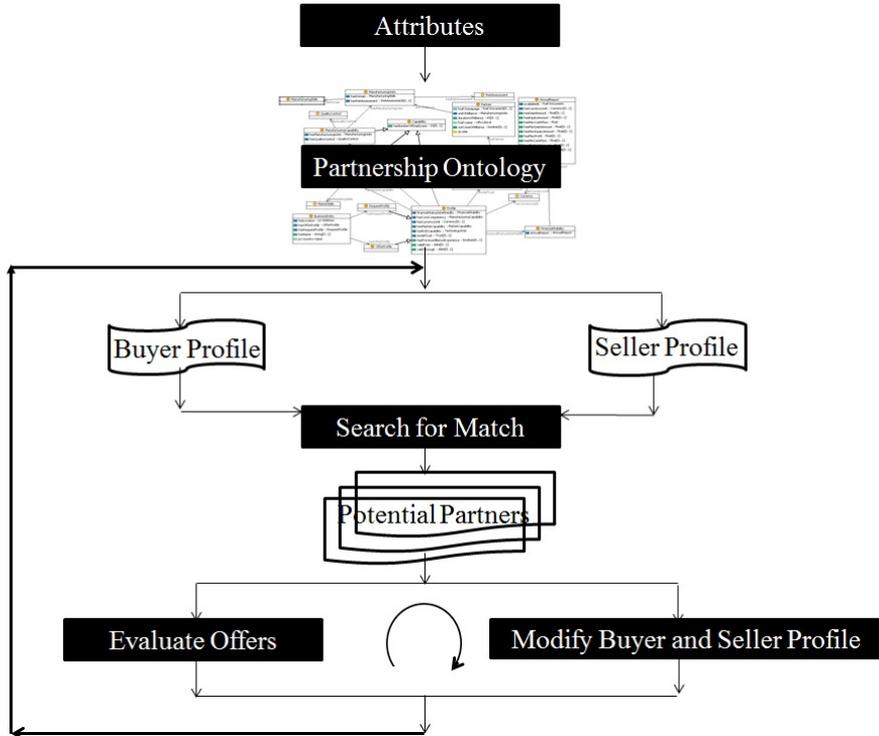


Figure 1.8: An archetype for Partnership match, showing the flow of processes

The architecture, shown in Figure 1.8, is developed in such a way that it prompts the user to adopt a systematic approach to partnership establishment. A web enabled software prototype is developed and used to validate the architecture. The success of partnerships establishment is significantly influenced by the manner in which profiles are created. A profile is simply a set of

1.6 Organization of this Dissertation

generic facts about a user requirements representing an enterprise, which may be used by other users to determine their suitability as potential partners. A seller profile records the capabilities and capacity of the potential partner. A buyer profile is a mechanism utilized to communicate to the SME what the potential partner can do to meet their needs. The first step of any partnership establishment process should take place with both the parties defining their terms (requirements and offer attributes). A user looking for SME partners makes a buyer profile; whereas, SMEs make a seller profile, note that both are oblivious of each other, i.e., they just make their profiles available to the system. Buyer, after providing his profile, searches for the matching seller profiles, which the system recommends after executing a semantic similarity match among various profiles available to the system. The result from searching is a set of possible partners that a buyer can consider to be his/her future partners. At this stage a buyer communicates his interest to the potential SME partners and negotiates by modifying his profile. In other words, profiling is acting as a communication channel between users. The next and final step is to select one of the SME partners from the list of available partners after negotiations and proceed with face to face meetings, discussing contract details, etc.

1.6 Organization of this Dissertation

To start with, so as to put the contributions in perspective, the Chapter 2 presents a through survey on relevant research topics. The topics include search

1.6 Organization of this Dissertation

engines, user profiling, matrix factorization, clustering, ranking algorithms, and related folksonomy based personalized search algorithms. The main contribution of this dissertation start with Chapter 3. In this chapter, a novel approach to construct a user profile, called as User Interest Profile (*UIP*) in this dissertation, from user interactions with the web is presented. It capitalizes on the user's search history and link structure of the web that includes anchor tags to build a *UIP* and use that for personalized search. In the next chapter, chapter 4, I explore folksonomy based approaches to construct *UIP* and *CUIP*. Two methods are presented that leverage upon the folksonomy to build a profile of user interests, called as *UIP*. The *UIP* is further processed using matrix factorization algorithms to extract hidden semantics in it so as to group related tags together that could be either syntactically related, semantically related, or contextually related. To group these related tags together into clusters, thus generating a Clustered User Interest Profile (*CUIP*), where each cluster identifies a unified topic, clustering algorithms are used. For the non-folksonomy based approach, one custom data set is used, and it compared the proposed method with other non-folksonomy based methods. Two different data-sets were constructed for the evaluation of folksonomy based methods for personalized search: twitter data-set and AOL query log . The twitter data set was established to evaluate the sparsity of information in *UIPs* and *CUIPs* and to test the clustering tendency and clustering accuracy of *CUIPs*; AOL data-set, which is a much larger data-set of user search histories, was harvested from AOL Search Query Log. This data set was used to test the improvement in

1.6 Organization of this Dissertation

personalized search for the two proposed folksonomy based methods and to compare them with other folksonomy based personalized search methods. In Chapter 5, I propose a partnership ontology that is used for building buyer profiles and seller profiles. A web service is developed that can be used by users for representing their respective profiles, and it also allows to find matching seller profiles for a given buyer profile. I conclude in Chapter 6 with summarizing remarks, a discussion on directions that the presented research topics can take here-on, and a general discussion on the future work.

2

Background

*The warrior who trusts his path doesn't need to prove the other is wrong. -
Paul Coelho*

We have witnessed great interest and a wealth of promise in content-based document retrieval as an emerging technology in the last decade. While a firm foundation has been laid, it also paved the way for a large number of new techniques and systems, got many new people involved, and triggered stronger association of weekly related fields. In this chapter, we survey key theoretical and empirical contributions in the current decade related to Social Semantic Web, Search Systems, User Profiling, Personalization, and Partnership Match.

2.1 Introduction to Social Web

The Social Web is an ecosystem of participation, where value is created by the aggregation of many individual user contributions (Tom, 2008). The Social Web is represented by a class of web sites and applications in which user

2.1 Introduction to Social Web

participation is the primary driver of value. The architecture of such systems is well described by Tim O'Reilly (Tim, 2005), who has fostered a community and media phenomenon around the banner of Web 2.0. Headliners for the festival include Wikipedia, MySpace, YouTube, Flickr, Del.icio.us, Facebook, and Technorati. Discussions of the Social Web often use the phrase "collective intelligence" or "wisdom of crowds" to refer to the value created by the collective contributions of all these people writing articles for Wikipedia, sharing tagged photos on Flickr, sharing bookmarks on Del.icio.us, or streaming their personal blogs into the open seas of the blogosphere. Tagging has become a valuable feature for organizing such resources. The potential for knowledge sharing today is unmatched in history. Never before have so many creative and knowledgeable people been connected by such an efficient, universal network. The costs of gathering and computing over their contributions have come down to the point where new companies with very modest budgets provide innovative new services to millions of on-line participants. The result today is incredible breadth of information and diversity of perspective, and a culture of mass participation that sustains a fountain of publicly available content.

Collective intelligence is a grand vision, one to which I subscribe. However, I would call the current state of the Social Web something else: collected intelligence. That is, the value of these user contributions is in their being collected together and aggregated into community- or domain-specific sites: Flickr for photos, YouTube for videos, etc. I think it premature to apply term collective intelligence to these systems because there is no emergence of truly new levels

2.1 Introduction to Social Web

of understanding. From the Social Web collective we can learn which terms are popular for tagging photos or the buzz in the latest blog posts, and we can discover the latest new talent in video, photography, or op-ed. However, while popularity is one measure of quality, it is not a measure of veracity. Mass authoring is not the same thing as mass authority. Particularly in the presence of spam and other fraudulent sources in the mix, simply collecting the contributions of the masses does not lead to new levels of intelligence.

Collective intelligence has been the goal of visionaries throughout the history of the Internet. Douglas Engelbart, who invented groupware, the mouse, and a form of hypertext designed for collective knowledge, wrote in 1963 of his career and project objective: "The grand challenge is to boost the collective IQ of organizations and of society" (Engelbart, 1962). His Bootstrap Principle was about a human-machine system for simultaneously harvesting the collected knowledge for learning and evolving our technology for collective learning. In human-machine systems, both the human and machine contribute actively to the resulting intelligence, each doing what they do best. Other early pioneers of the human-machine model of collective intelligence include Norbert Wiener, the father of cybernetics, Buckminster Fuller, the consummate inventor and system thinker, and Stewart Brand, creator of the first large virtual community on the Internet (Fred, 2006).

The key, as the visionaries have seen, is a synergy between human and machines. Clearly, there are different roles for people and machines. People are the producers and customers: they are the source of knowledge, and they have

real world problems and interests. Machines are the enablers: they store and remember data, search and combine data, and draw mathematical and logical inferences. People learn by communicating with each other, and often create new knowledge in the context of conversation. The Internet makes it possible for machines to help people create more knowledge and learn from each other more effectively. With the rise of the Social Web, we now have millions of humans offering their knowledge on-line, which means that the information is stored, searchable, and easily shared. The challenge for the next generation of the Social and Semantic Webs is to find the right match between what is put on-line and methods for doing useful reasoning with the data. True collective intelligence can emerge if the data collected from all those people is aggregated and recombined to create new knowledge and new ways of learning that individual humans cannot do by themselves.

The Social Web reflects that more and more Web systems accomplish an architecture of participation, which involves participation of end-users. Resource sharing systems like Flickr or YouTube depend on their users, who contribute pictures and videos, because the main purpose of these systems relies in sharing user-contributed content. Social tagging supports resource sharing within these systems (Hotho, Jäschke, Schmitz, and Stumme, 2006): "social resource sharing systems are Web-based systems that allow users to upload their resources, and to label them with arbitrary words, so-called tags". For example, in Flickr a user may publishes pictures from her latest travel to France, which she annotates with keywords such as *france*, *Paris* or *beautiful-nature*. These

tags will help the user to retrieve certain images in the future and therewith support her and others, as we capitalize in this work, personal information management (Heckner, Heilemann, and Wolff, 2009). Further, other users will be enabled to find the pictures if they utilize the corresponding tags to search for Flickr pictures (Kumar and Kim, 2012, 2011, Lee, Kim, Shin, and Kim, 2009, Sigurbjörnsson and Van, 2008).

Social tagging does not require pre-defined taxonomies, but vocabularies used for organizing resources in tagging systems rather emerge like desire lines (Schmitz, 2006). The structures that emerge from social tagging are called folksonomies. The term folksonomy was first introduced by Thomas Vander Wal (Vander, 2005, Feb 2007) and depicts the structures that evolve over time when users (the folks) annotate resources with freely chosen keywords. Folksonomies relate users, tags and resources based on the tag assignments that are performed by the user community. Tag assignments are triples that state which user assigned which tag to which resource. Hence, a folksonomy can thus be considered as a collection of tag assignments and folksonomy systems are those systems that allow for the evolution of folksonomies.

Today, there exist many diverse folksonomy systems in various domains. For example, Last.fm enables users to annotate music, bookmarks can be tagged in systems such as Delicious, BibSonomy supports social tagging of research articles, Amazon enables their customers to tag products, and Google Mail users can organize their emails via freely chosen labels.

2.2 Matrix Decomposition Methods

Data Mining is about finding new and interesting information from data (Jiawei and Micheline, 2001). The underlying assumption is that there is too much data for a human to process, and thus one needs an automated method that can process the corpus and find interesting and relevant information. Given the huge amount of data, it is computationally time consuming job to execute data mining or machine learning algorithms on them. Matrix decomposition methods are executed as a pre-processing step where the objective is to filter out less relevant information and only keep the more relevant ones.

Matrix decomposition, where a given matrix is represented as a product of two or more matrices, are regularly used in data mining. Most matrix decompositions have their roots in linear algebra, but the needs of data mining are not always those of linear algebra. One of the basic concept of Matrix decomposition algorithms is a matrix. In linear algebra, an n -by- m matrix is usually interpreted as a linear map from n -dimensional space to m -dimensional space (Gene and Charles, 1996). But, in data mining, and also in this dissertation, matrices are a convenient way to store and manipulate data. We have used matrices for storing text documents as term frequency matrices (Jiawei and Micheline, 2001).

Every matrix decomposition has three concepts related to it. First of these is the formulation of decomposition, that is, to what kind of matrices the decomposition applies (example, only to non-negative matrices or only to binary

2.2 Matrix Decomposition Methods

matrices), and what kind of factor matrices are feasible for the decomposition (example, non-negative matrices or orthogonal matrices). Second concept is the concrete decomposition of some matrix \mathbf{A} . Third concept is the problem of finding a decomposition that admits the formulation, given some matrix \mathbf{A} . When performing a matrix decomposition on some matrix, it is represented as a product of two or more factor matrices. The most widely used method to decompose a matrix is the Singular Value Decomposition (SVD)(Gene and Charles, 1996). It decomposes a matrix \mathbf{A} into the form $U \Sigma V^T$, where U and V are orthogonal matrices, that is $U^T U = V^T V = \mathbf{I}$, and Σ is a diagonal matrix with non-negative entries - the singular values of \mathbf{A} . The Singular Value Decomposition gives the optimal rank- k approximation of the original matrix \mathbf{A} . The optimal rank- k approximation of \mathbf{A} can be obtained from its Singular Value Decomposition by setting all of the k largest singular values to 0. Computing the SVD is also relatively fast; it can be done in time $O(\min n^2 m, n^2 m)$ for n -by- m matrices (Gene and Charles, 1996). The methods often employed in practice, such as Lanczos methods (Gene and Charles, 1996), are usually even faster. Nevertheless, for extremely large matrices that can still be overwhelming. This has motivated the study of fast, approximate decomposition algorithms that are based on sampling the original matrix. Work done in this field include the results of (Alan, Ravi, and Santosh, 2004), (Drineas, Kannan, and Mahoney, 2006a), (Drineas, Kannan, and Mahoney, 2006b), (Drineas, Mahoney, and Muthukrishnan, 2006c), (Drineas, Mahoney, and Muthukrishnan, 2006d), (Drineas, Mahoney, and Muthukrishnan, 2008), and (Achlioptas and

2.2 Matrix Decomposition Methods

McSherry, 2001).

If a matrix A is non-negative (example, because it is a result of measurements that can only yield non-negative results), interpreting the results of SVD can be problematic. This is because for a non-negative matrix A , the U and V factor matrices produced by SVD can contain non-negative values. This problem is addressed by Non-negative Matrix Factorization (NMF) methods, where the factor matrices are required to have only non-negative values. Early formulations of the NMF problem include (Paatero and Tapper, 1994), where they called it ‘positive matrix factorization’, and (Cohen and Rothblum, 1993). However, the most famous is due to (Lee and Seung, 1999). Since their article, the problem has attracted a lot of attention and many researchers developed an innumerable number of algorithms (Berry, Browne, Langville, Pauca, and Plemmons, 2007).

In addition to SVD and NMF, many other matrix decomposition algorithms have been proposed, most of which are based on probabilistic models. Such methods include multinomial Principal Component Analysis (MPCA) (Buntine, 2002), probabilistic Latent Semantic Indexing (PLSI) (Hofmann, 1999a,b, 2001, Papadimitriou, Tamaki, Raghavan, and Vempala, 1998), and Latent Dirichlet Allocation (LDA) (Blei, Ng, and Jordan, 2003). There has been some research on expressing these decompositions in a unified way ((Buntine and Jakulin, 2006) and (Singh and Gordon, 2008)).

2.3 User Interest Profile For Personalized Web Search - Non Folksonomy based

Google's innovative page ranking search (Brin and Page, 1998) revolutionized the use of SEs. PageRank uses the citation graph of the Web along with the introduction of link analysis in SE systems. SEs, such as Google, Yahoo, and MSN, do a commendable job for experienced users, but fail to satisfy the needs of naive users. (Teevan, Dumais, and Horvitz, 2007) reported that although SEs provide the best possible result set, they are not satisfactory at individual user levels. Search results can be improved by personalization (Chirita, Firan, and Nejdl, 2006, Eugene, Eric, Susan, and Robert, 2006, Kelly and Teevan, 2003, Ma, Pant, and Sheng, 2007, Shen, Tan, and Zhai, 2005, Teevan, Dumais, and Horvitz, 2005), by, for example, recommending varying results to different users for the same query. The results are differentiated based on user interests, which are obtained from the user's *UIP*. Automatic construction of a *UIP* usually deals with the observation of user browsing behaviour. (Kelly and Teevan, 2003) reviewed several possible approaches to inferring user preferences by categorizing user behaviour across many dimensions such as examine, retain, and reference. (Agichtein, Brill, and Dumais, 2006) organized user interests as a set of features that are organized into three groups: Query-text, Click-through, and Browsing. The Query-text feature includes result title, URL, and summary. The Click-through feature includes Click-Frequency (number of clicks for the result), IsClickBelow (whether there was a click on a result below

2.3 User Interest Profile For Personalized Web Search - Non Folksonomy based

the current URL), and IsClickAbove (whether there was a click on a result above the current URL). The Browsing feature includes TimeOnPage, TimeOnDomain, and the deviation of the dwell time from the expected dwell time for a query. (Shen, Tan, and Zhai, 2005) collected user interests from clicked document summaries, titles, Click-Through histories, and query histories that were accumulated over a session. (Teevan, Dumais, and Horvitz, 2005) and (Chirita, Firan, and Nejdl, 2006) used the files on the user's desktop to construct a *UIP*. A major limitation of these approaches is that there can be a lot of terms on the user's desktop, which makes a *UIP* noisy or misleading. (Das, Datar, Garg, and Rajaram, 2007) used collaborative filtering (CF) for personalization. The underlying assumption of the CF approach is that users who agreed in the past tend to agree again in the future.

A rather simplistic approach to construct a *UIP* is to explicitly ask a user for his/her topics of interest. The *UIP* is then used for filtering search results by checking content similarity between the returned Web documents and the *UIP*. Early versions of Google personalization asked the user to choose the categories of interest. The Google SE applied this information to filter search results. An inherent limitation of this approach is that user interests are subject to changes over time. Moreover, (Carroll and Rosson, 1987) showed that users are reluctant to provide explicit information about their interests or any explicit feedback on search results. Other important methods using ontologies emerged as well in which a *UIP* is constructed by classifying Web pages in the user's web browser cache into appropriate concepts in the reference ontology

2.4 User Interest Profile for Personalized Web Search - Folksonomy based

(Gauch, Chaffee, and Pretschner, 2003) and (Speretta and Gauch, 2005a) or ODP (Chirita, Firan, and Nejd, 2006).

2.4 User Interest Profile for Personalized Web Search - Folksonomy based

Recently, some research works have investigated social bookmarking services for building and applying a *UIP* for personalized search (David, Iván, and Joemon, 2010, Kumar and Kim, 2012, 2011, Noll and Meinel, 2007, Xu, Bao, Fei, Su, and Yu, 2008) and resource recommendation (Abel, Gao, Houben, and Tao, 2011, Andriy, Jonathan, Bamshad, and Robin, 2008, Fabian, Nicola, Eelco, and Daniel, 2010, Vallet and Castells, 2012).

The approaches by (Noll and Meinel, 2007), (Xu, Bao, Fei, Su, and Yu, 2008), and (David, Iván, and Joemon, 2010) for personalized search builds a *UIP* from the tags that the user uses to annotate resources. A Resource Profile(*RP*) for a resource is constructed from the tags that the community has used to annotate it. A resource clicked by a user manifests the user's interest in it and possibly the tags associated with it. Tags assigned by a user to a resource can hardly be a complete description of the resource. However, collective tagging of a resource by a community of users provides a more complete description of it. We believe that there are syntactical differences between the search terms that a user uses and the terms found in a search result document. Each user has a specific vocabulary of terms that he/she uses to formulate a query. And

2.4 User Interest Profile for Personalized Web Search - Folksonomy based

Table 2.1: A comparison summary of the proposed approaches with the other similar approaches that uses folksonomy for personalized search. (a)Source of terms for constructing a *UIP*, (b) Web document Representation, (c) Similarity Measure, (d)First-Order Co-occurrence, (e) Second-Order Co-occurrence, (f)Clustering of terms in a *UIP*, (g) *UIP* and resource length normalization factor

	<i>tfUIP</i> Noll and Meinel	<i>tfIdfUIP</i> Xu et al.	<i>tfIdfCUIP</i> Andriy et al.	<i>svdCUIP</i>	<i>modSvdCUIP</i>
(a)	User annotations	User annotations	User annotations	Annotations by the community	Annotations by the community
(b)	Resource Profile (folksonomy based)	Resource Profile (folksonomy based)	Resource Profile (folksonomy based)	document contents	document contents
(c)	dimensionless cosine similarity measure, Equation 2.1	tfIdf cosine similarity between a <i>UIP</i> and a <i>RP</i> , Equation 2.2	cosine similarity between a global cluster matching a user query and the <i>RP</i>	cosine similarity between the matching cluster in the <i>CUIP</i> to the user query and the document contents, Equation 4.6.	calculates the cosine similarity between the matching cluster in the <i>CUIP</i> to the user query and the document contents, Equation 4.6.
(d)	Yes	Yes	Yes	Yes	Yes
(e)	No	No	No	No	Yes
(f)	No	No	Yes	Yes	Yes
(g)	No	Yes	Yes	Yes	Yes

each author of a document has his/her own vocabulary of terms too. Chances are that the vocabularies are different. The rift effectively results in the low similarity score or re-ranking score between the search result and the *UIP*. Note also that there can exist similarity in semantics among the terms in the user's *UIP* and the *RP* of the result document. If a *UIP* consists of all the tags, used by a community of users, to annotate the resources of user interests, it is very likely to have a greater correspondence between the *UIP* and the *RPs* of result documents. Hence, it is our proposal that a *UIP* should consist of all the tags used by a community of users to annotate the documents or resources clicked by the user. We have adapted the approaches presented in (Noll and Meinel, 2007), (Andriy, Jonathan, Bamshad, and Robin, 2008), and (Xu, Bao, Fei, Su, and Yu, 2008) to construct a *UIP* by amalgamation of tags from the *RPs* of the resources or Web documents clicked by the user. We are of the opinion that any of these approaches can be benefited by the application of SVD, an approach proposed by us to construct a *CUIP*.

2.5 Personalized Search

One of the issues with personalized search is how to acquire the index? The construction of an index is a tedious process. An alternative option is to use the search results available from the SE. Most SEs do not allow scrapping of search results. However, they do provide search APIs with limited access and some restrictions. Researchers use Google API, Yahoo API, or MSN API to retrieve

search results. (Kumar and Kang, 2008) provided a comprehensive look at the differences in search results obtained from a SE and a SE API for the same input query, i.e., How well a SE API surrogates a SE? The following differences are reported: freshness, accuracy, ranking, the number of results, and the difference in index. They reported that Yahoo SE and Yahoo search API have same search quality, this is to say, underneath both use the same index, unlike Google API and MSN API use a different index than used by Google and MSN, respectively. This work uses Google API for retrieving search results.

(Pitkow, Schütze, Cass, Cooley, Turnbull, Edmonds, Adar, and Breuel, 2002) described two approaches to personalizing Web search results: query expansion (Chirita, Firan, and Nejd, 2006, Gauch, Chaffee, and Pretschner, 2003, Speretta and Gauch, 2005a) and re-ranking of search results (David, Iván, and Joemon, 2010, Ferragina and Gulli, 2005, Koshman, Spink, and Jansen, 2006, Noll and Meinel, 2007, Wang and Jin, 2010). In query expansion, user interests are conflated with a given query, and the expanded query is used for searching the Web. For re-ranking of search results, the SE results are re-ranked by computing the similarity between the document contents and the terms in the UIP.

(Agichtein, Brill, and Dumais, 2006) used supervised machine learning technique, named RankNet, for re-ranking search results. (Dou, Song, and Wen, 2007) used S.E logs for constructing user profiles. Further they re-rank search results by computing a personalized score for each URL in the result set. They introduced four formulas for re-ranking: two methods closely relate to collab-

orative filtering, and the other two relate to personal level. (Ferragina and Gulli, 2005) proposed web snippet clustering, in which the search results are presented hierarchically using web snippets. It clusters snippets returned by a SE into a hierarchy of folders which are labelled with variable length sentences. The labels are named such that they represent the theme of the snippets contained into their associated folders. For personalization, users can select a set of labels, and ask the SE to filter out all other labels except the selected ones. Note that their approach is bounded towards clustering search results, whereas our approach is bounded towards clustering terms to generate a CUIP and using the CUIP for personalized search.

The method by (Noll and Meinel, 2007), referred to as *tfUIP* in this thesis, re-ranks a document by computing the dimensionless cosine similarity between the tags in the *RP* of the document and the *UIP*.

$$tfUIP(UIP, d) = \sum_{t \in UIP, tf_d(t) > 0} tf_{UIP}(t) \quad (2.1)$$

The method by (Xu, Bao, Fei, Su, and Yu, 2008), referred to as *tfIdfUIP*, re-ranks a document by computing the cosine similarity between the tags in the *RP* of the document and the terms in the *UIP*.

$$tfIdfUIP(UIP, d) = \frac{\sum_t (tf_{UIP}(t) \cdot idf_{UIP}(t) \cdot tf_d(t) \cdot idf_d(t))}{\sqrt{\sum_t (tf_{UIP}(t) \times idf_{UIP}(t))^2} \cdot \sqrt{\sum_t (tf_d(t) \times idf_d(t))^2}} \quad (2.2)$$

The method by (David, Iván, and Joemon, 2010), an adapted approach of (Xu, Bao, Fei, Su, and Yu, 2008), referred to as *tf-iuf* in our work, excludes length normalization factors of the *UIP* and documents from the similarity score computation, and includes the inverse user frequency and inverse document frequency.

$$tf - iuf(UIP, d) = \sum_t (tf_{UIP}(t) \cdot iuf_{UIP}(t) \cdot tf_d(t) \cdot idf_d(t)) \quad (2.3)$$

The justification for exclusion of document length normalization factor is similar to that of *tfUIP* that using the document length normalization factor would penalize the score of popular documents. The reason for exclusion of *UIP* length normalization factor is that in all computations of similarity scores, the *UIP* length normalization factor is constant. Similar to *tfUIP*, *tfIdfUIP* and *tf-iuf* use all terms in the user’s *UIP* for computation of similarity scores to re-rank search result documents.

Recent work(Bouadjenek, Hacid, and Bouzeghoub, 2013a, Bouadjenek, Hacid, Bouzeghoub, and Vakali, 2013b) on folksonomy based personalized search builds a personal document representation (PSDR) in a social collaborative setting. Further, a ranking function is proposed to rank documents using PSDR.

The method by (Andriy, Jonathan, Bamshad, and Robin, 2008) presented a personalization algorithm for recommendation in folksonomies, referred to as *tfIdfCUIP* in our work, which relies on hierarchical tag clusters. Their approach clusters the entire tag space of a folksonomy system to obtain one common,

global cluster structure available to those users who are registered with the folksonomy system. This restrains the outreach of the approach. Further, they gauge user interest in each tag cluster based on the user usage of tags for resources' annotations. A set of matching clusters extracted from the overall clustered tag space makes up a *CUIP* to be used for personalized resource recommendation. And, both tf-idf and tf are used to compute the similarity score of resources and a *CUIP*.

Our proposed methods, for personalized search based on *svdCUIP* and *modSvdCUIP*, use a *UIP* length normalization factor during similarity score computation because the methods expand the user query with the tags from the matching cluster in the user *CUIP*, and compute the similarity score between the expanded query and the document contents. The *UIP* length normalization factor varies in accordance with queries because each query may match to a different tag cluster. Because *RPs* can only be constructed for a small subset of documents, we refrain from using *RPs* of documents for ranking them. The methods calculate the similarity between the expanded query and document contents. In fact, we have found that it is only possible to construct *RPs* for approximately 50% of Web documents when using social bookmarking services. This seriously jeopardizes the outreach or acceptability of personalized search systems.

In a nutshell, the *tfUIP* and *tfIdfUIP* re-rank the search result set by computing the similarity scores between the terms in the *UIP* and *RPs* of documents in the result set, whereas the proposed approaches are based on query expansion and use document contents for ranking search results.

2.6 Partnership Match

Existing work in the domain of Partnership Match is focused towards total ranking of the Suppliers (Chen, Lin, and Huang, 2006, Chen, Lee, and Wu, 2008, Dulmin and Mininno, 2003, Lin, Xu, and Xu, 2010, Liu and Hai, 2005, Sun, Ji, and Xu, 2009). Hence, these works provide some sort of weight procedure based on ANP (Chen, Lee, and Wu, 2008), Fuzzy Logic(Chen, Lin, and Huang, 2006), Data Mining(Lin, Xu, and Xu, 2010), and AHP (Liu and Hai, 2005). The research work related to Partnership Match can be classified into following categories: AI Systems (Chen, Lin, and Huang, 2006, Liu and Hai, 2005), Mathematical Models (Chen, Lee, and Wu, 2008, Choy and Lee, 2003, Dulmin and Mininno, 2003), Ontology Models(Li, Wu, and Yang, 2004a, Li, Huang, Liu, Gou, and Wu, 2001), Statistical(Petersen and Divitini, 2002), and Simulation Studies(Basnet and Leung, 2005, Cakravastia and Takahashi*, 2004). We place our work under Ontological models.

(Chen et al., 2006)propose to solve supplier selection or partnership establishment problem by building a hypothesis that there is an uncertainty involved in decision variables of partner attributes. Therefore, they propose to use fuzzy algorithms. However, their work is based on preliminary screening, which means, the process is partially automatic. Most of the research work in this area revolves around using Mathematical Models(Chen, Lin, and Huang, 2006, Choy and Lee, 2003, Dulmin and Mininno, 2003, Min, 1994). Some authors formulated the partnership establishment problem as Analytic Network

2.6 Partnership Match

Process(Bayazit, 2006, Chen, Lin, and Huang, 2006), some use Case Based Reasoning(Choy and Lee, 2003), and Multi Attribute Utility tool(Min, 1994, Sun, Ji, and Xu, 2009). Interestingly, an organization profile which consists of quantitative attributes and qualitative attributes has to be modeled such that they can be effectively used for numerical calculation(Dickson, 1966). The problem arises when modeling qualitative attributes for numerical calculation- solution to which is often provided by using mathematical Models. The qualitative features modeled use a scale indicating the strength with which one factor dominates another with respect to a higher level factor. However, in the aforementioned research work, the list of attributes to model a profile is not comprehensive, for ex: (Choy and Lee, 2003) and (Dulmin and Mininno, 2003) fail to take into account marketing capabilities, financial stabilities, and cultural alignment etc. We have tried to cover all the features for modeling a profile. This enforces the fact that different companies have different specific requirements concerning supplier evaluation. For instance, (Schmitz and Platts, 2004) used a semi-structured questionnaire in several European locations to collect opinions and suggestions from automobile suppliers on vendor performance evaluation. One of the key results of their study is that the evaluation of supplier includes management information, communication, motivation of suppliers, coordination and alignment, decision making and priority learning. A number of simulation studies with a focus on the partner establishment have also been published. (Crama et al., 2004) formulated a non-linear 0-1 programming problem with complex quantity discounts offered by different suppliers

2.6 Partnership Match

and alternative product recipes. (Cakravastia and Takahashi*, 2004) created a simulation model to determine which supplier to select for business and the volume assigned to each of those suppliers. Finally, (Basnet and Leung, 2005) created a simulation model to determine what products to order in which quantities from which supplier in which periods to satisfy a given demand stream. One major task of purchasing manager is selecting the right supplier. Suppliers have varied strengths and weakness which requires meticulous evaluation by the purchasing manager before ranking them. The foremost task is to establish the criteria or features for supplier evaluation. (Weber et al., 1991) classified 74 articles, on the 23 criteria from (Dickson, 1966), related to supplier selection and discussed the effect of various features on supplier selection. Since different enterprises have different requirements in terms of supplier evaluation, i.e., they use different set of features therefore in this work, we have arranged a comprehensive list of features required by purchase managers and further represented those features as an ontology. Ontologies are the structural frameworks for organizing information and are used in Grid Computing(Lee, Lee, Noh, and Han, 2010) (Jang, Lee, Noh and Han 2010), WWW (Sui and Zhao, 2009), systems engineering (Pham and Jung, 2010), etc, as a form of knowledge representation about the world or some part of it. (Li et al., 2001) and (Li et al., 2004a) use ontology for modeling partner profile; however, authors fail to provide any case study that can demonstrate their work. (Petersen and Divitini, 2002) use statistical model for calculating similarity between partners. Their model particularly works for software projects; hence the features for modelling

2.6 Partnership Match

profiles are more technically oriented rather than being generalized. It uses an agent oriented approach and Multi-Attribute Utility Function to determine the score of partners which is further used for ranking. This work suffers from the drawback that it only works for virtual enterprise that is formed for a software project.

3

Mining anchor text for building User Interest Profile: A non-folksonomy based personalized search

The very first search engine was developed by Gerard Salton, and it was called the SMART information retrieval system(Salton, 1971). The first pre-web search engine was Archie(Van Couvering, 2008), which allowed searching for file names of a database. The early search engines retrieved results from their indexed database and displayed the cached pages based on keyword match and similarity measures. Traditional indexing methods worked quite well for database or structured information but later it was discovered that they are not

compatible for indexing unstructured information such as World Wide Web. The search engines based on simple indexing technologies were Lycos, Alta Vista etc. (Brin and Page, 1998) proposed an innovative page ranking system which revolutionized the use of search engines. Page rank uses the citation graph of the web and Google introduced link analysis in the search engine systems.

To improve the quality of search results returned by a search engine, many solutions have been proposed: first is to use a Vertical Search Engine (Koshman, Spink, and Jansen, 2006) for specific information needs, second is the use of a personalized search engine (Chirita, Firan, and Nejd, 2006, Das, Datar, Garg, and Rajaram, 2007, Ferragina and Gulli, 2005, Gauch, Chaffee, and Pretschner, 2003, Speretta and Gauch, 2005b, Sun, Zeng, Liu, Lu, and Chen, 2005, Teevan, Dumais, and Horvitz, 2005, 2007), and third is to improve search engine results (Chakrabarti, Dom, Gibson, Kumar, Raghavan, Rajagopalan, and Tomkins, 1998a, Chakrabarti, Dom, Raghavan, Rajagopalan, Gibson, and Kleinberg, 1998b, Haveliwala, 2002). Personalized Search has emerged as an effective solution to improve quality of search results. Using Topic Distillation (Chakrabarti, Dom, Gibson, Kumar, Raghavan, Rajagopalan, and Tomkins, 1998a) and ARC (Chakrabarti, Dom, Raghavan, Rajagopalan, Gibson, and Kleinberg, 1998b), Chakrabarti et al. has showed how quality of search results can be improved. Another similar attempt by Haveliwala (2002) used hub vectors limited to 16 for calculation of topic sensitive page rank. These approaches will improve search results but they do not provide different results for different

users. Using a Vertical Search Engine is not appropriate in all cases as they have an inherent restriction that they are restricted to one specific domain.

I want to leverage upon feature based user profiling, refer Figure 1.1, for building a profile of user interests. In this chapter, a profile of user interests is built from the anchor text of the clicked Web pages in the user search history. This type of method is called as non-folksonomy based method for building a profile of user interests. The anchor text represents the feature that is being mined to represent user interests. In chapter 5, I propose a more advanced method to build a profile of user interests that uses a different feature. Both non-folksonomy and folksonomy based methods in chapter 4 and 5 are used for personalized search. In chapter 6, a practical approach to build a use profile from explicit user involvement is presented, and the user profiles are used for partnership match.

This chapter makes the following contributions:

1. I propose a non-folksonomy based personalized search method, 'Exclusively Yours', that capitalizes on the anchor text to construct a User Interest Profile (*UIP*).
2. I propose a term-weighting method specifically targeted to this work with the goal of accumulating weight of terms emanating from the linked Web pages of clicked documents.
3. I also propose a model to logically segregate a *UIP* into two parts based on the latency of terms in the *UIP*. It effectively discounts term weight

of those terms in the *UIP* that have not been updated over a period of time.

4. The proposed method is compared with the other non-folksonomy based personalized search methods and with the non-personalized Web search.

To achieve good personalization (Ferragina and Gulli, 2005), three requirements have been stated: full adaptivity to the changing user behaviours/needs, privacy protection, and scalability. Our proposed method satisfies all the three aforementioned requirements. To take care of user behaviour needs that may change over a period of time, we construct two types of user profile, p_{perm} and p_{temp} . Regarding the privacy, we make no attempt to infringe in user personal data or personal files as has been done by few personalization techniques (Chirita, Firan, and Nejd, 2006, Teevan, Dumais, and Horvitz, 2005). Regarding scalability, we tested our system for many months and with many users, the results obtained were satisfactory.

3.1 Exclusively Yours'

Figure 3.1 is an illustration of Exclusively Yours' system architecture that provides personalized search results for a given user query. On the client side, a user requests a query and chooses a search engine from the available four options (Google, Yahoo, MSN, and Naver). The retrieved search results (a set of ranked URLs) are logged along with the query and the user ID. Each user is supposed to register before he/she can use the proposed system. Each

3.1 Exclusively Yours'

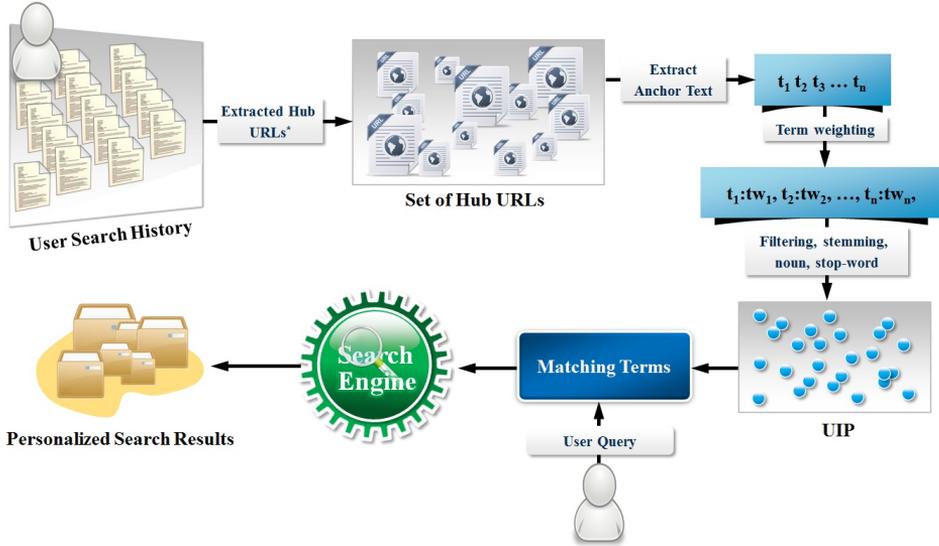


Figure 3.1: System Architecture of Exclusively Yours'

user logs in using his unique user ID. The user ID and other information are logged. The logged information is used during experiment for identification of a session. If a user clicks or downloads a URL, the system logs the selected URL along with the query and the user ID. The anchor text extraction module extracts anchor text and its surrounding text from the associated hubs of each URL clicked or downloaded by the user. We have proposed a weighting scheme that assigns weight to each extracted term. The weight is computed in the weight computation module. The weight assigned is based on the rank of URL and the rank of associated hub URL that contains the anchor text. Moreover, the extracted terms are stored in an indexed file along with their weights, and

various other attributes. The User Interest Profile (*UIP*) consists of extracted terms which will be used later for expanding user query.

3.1.1 Infer User Interests

This section describes our approach and the experiments that we use to set values for the small number of parameters in the algorithm. We have divided the whole process into three phases: 'training' phase, 'weighting' phase, and 'testing' phase. Given a query q , let U be the set of URLs returned from a user selected search engine which can be Google, Yahoo, or Naver (a Korean Search Engine). Let V ($V \subset U$) is the set of URLs clicked or downloaded by the user as shown in Figure 3.2. We now propose two fundamental ideas. The first idea,

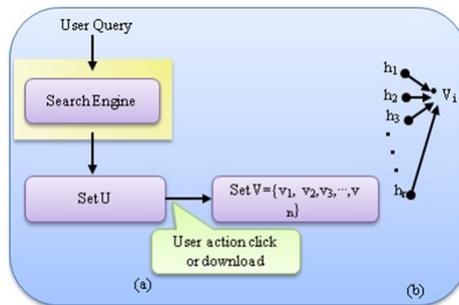


Figure 3.2: Set U represents URLs returned by a search engine and set V represents URLs clicked or downloaded by the user. On the right (b), URLs $h_1, h_2, h_3, \dots, h_n$ are hub URLs for URL V_i .

we need to find user interests using hyperlink structure. The second idea which is explained in detail in section 3.1.2 expands user query by conflating it with

a set of related terms from the *UIP*. To achieve the goal of first idea, hub pages are determined for the web pages in set V : for each URL v_i in the set V , find the top n web pages that are hub pages of v_i , i.e., web pages that have a link to a page v_i , as shown in Figure 3.2. If a page u has a link to a page v then u is a hub page for a page v .

We believe that the URLs that a user clicks or downloads are related to his/her interests. It has been reported by (Kraft and Zien, 2004) that there exists similarity between search queries and anchor text. They also showed that anchor text is a succinct description of a web page. Therefore we extract anchor text and its surrounding text from the hub pages of URLs clicked or downloaded by the user to create an index file of extracted terms.

We are only interested in hub pages because it gives a comprehensive description of hyper-linked outgoing linked web pages. From each of the n hub pages corresponding to v_i , extract a window of size 50 bytes surrounding the anchor texts from an anchor tag that has a href (hypertext reference) link to page v_i . A similar work by McBryan (1994) has defined a window of size 50 bytes surrounding an anchor text as anchor window. To extract the text circumscribing the anchor text, the first step is to get rid of html tags around it. The following step is removal of stop words and stemming. The resulting text is indexed and assigned weight w_i . The process of calculating weight is explained in the next section.

Here is an example to demonstrate how anchor text and its surrounding text is extracted. For ex: a user entered a query *Hollands Opus*. The topmost result

Table 3.1: Top three Hub URLs for the IMDB URL

www.imdb.com/title/tt0113862/	
$HubURL_1$	www.math.harvard.edu/~knill/mathmovies/index.html
$HubURL_2$	www.salocin.com/weblog/archives/2004_04.html
$HubURL_3$	http://www.timlebon.com/wise-books/

Table 3.2: Terms extracted from the $HubURL_1$

www.math.harvard.edu/~knill/mathmovies/index.html				
Force	Choose	mozart	read	write
Long	Division	Cut	Art	Kid
Holland's Opus	Movie	Math		

is a URL (<http://www.imdb.com/title/tt0113862/>). This URL is provided by IMDB and provides the comprehensive information about the movie *Holland's Opus*. If the user clicks this URL, the hub URLs are extracted using the query (`link:http://www.imdb.com/title/tt0113862/`) directed to yahoo web service. As a result, the top three URLs that point to IMDB URL returned by yahoo service are presented in Table 3.1. The $HubURL_1$ has an anchor text that has HREF link to IMDB URL. On careful examination of this anchor text, we find out that most of the text surrounding the HREF was `<table>` tags. After removal of tags, the extracted text is *If I'm forced to choose f... to read and write about*. Interested readers can find the complete text by browsing $HubURL_1$. Table 3.2, shows terms returned after parsing, stop-word removal, and stemming. This is the final set of terms which is indexed with weight assigned to each one of them. Extracted terms from the k hub pages are indexed in a file called as index file. Each term in the index file is assigned

a weight. The procedure for weight calculation is explained in the next section. We believe that the text around HREF links to a page v_i is descriptive of the contents of v_i .

3.1.2 Weight Computation

One of our major contribution is the computation of the weight w_i for each extracted term. The idea is to assign the log of rank of the hub page that contains the anchor text to w_i as shown in equation 3.1 where R_{kj} is the rank of k^{th} hub page associated with the j^{th} URL clicked/downloaded by the user. Note that, R_{kj} is subtracted from the count of results in the first page.

$$w_i = \sum_{j=1}^{|V|} \sum_{k=1}^{|H|} \frac{\log R_{kj}}{\log R_j} \quad (3.1)$$

The denominator, R_j is the rank of j^{th} URL clicked by the user, acts as a parameter of penalization. It controls how much a rank at a lower position is penalized. Because $\log 1 = 0$, which will result equation 3.1 to infinity, instead we have used $\log 1 = 1$ for computation.

The parameter H represents a set of hub pages associated with the URL j . The double summation in equation 3.1 accumulate term weights if a term reoccurs in either or all the hub pages associated with the URL j . Further, if an extracted term appears in a web page that already exists in an index file, then its weight is cumulatively added. Also note that, the value of weight w_i is highly responsible for separating noise, i.e., those terms which do not correspond to user interest

will not occur too often and hence will have lower weight. Whereas the terms the occur too often will subsequently have higher weight thus indicating user interests. It can be argued that there will be a lot of such terms. We found out that there is indeed a lot of terms that represented user interests; these terms were also somehow related, for ex, from Table 3.1, one can see that movies, art, Mozart are closely related terms, they have high contextual similarity. To resolve the ambiguity, such contextually similar terms can be grouped together, i.e., those terms that are contextually similar are grouped together. One term which has highest weight can collectively represent such a group of terms. To determine the contextually similarity between terms, we have used Normalized Information Distance (NID)(Li, Chen, Li, Ma, and Vitányi, 2004b). The idea behind NID is that the terms that are closely related occur together in almost all the documents and hence their NID value evaluates to close to 1. For ex: if terms t_1 and t_2 are closely related, then the number of documents in which t_1 appears will be more or less similar to the number of documents in which t_2 appears. Those terms that are not closely related, have less frequency to occur together and their NID value is a larger number. Since user interests may change over a period of time, a *UIP* is logically viewed in two forms, p_{perm} and p_{temp} . The p_{perm} represents *UIP* for all days prior to current day and p_{temp} represents *UIP* for the current day. The *UIP* p_{temp} consists of terms collected for the current day and p_{perm} consists of terms collected during few days before current day. p_{temp} is constructed through the following process. We construct a vector a_t of terms collected from the hub pages corresponding to each web

page in V as follows:

$$a_t^i = \{a_{t_1}^i, a_{t_2}^i, a_{t_3}^i, \dots, a_{t_n}^i\} \quad (3.2)$$

where n is the number of terms extracted from URL v_i and its corresponding hub pages. The term at collected during one session is calculated as follows

$$a_t = \bigcup_{i=1}^{|V|} a_t^i \quad (3.3)$$

We divide user activity into various sessions during a particular day, i.e., each query represents one session. Moreover, we take union of all terms collected over all the m sessions in a single day which is represented as profile p as shown below.

$$p = \bigcup_{j=1}^m a_t^j \quad (3.4)$$

Finally, each term t_i is associated with two attributes; weight w_i and date of activity $a(t)$. The term date of activity is defined as the date when the weight of term was last updated. As shown in the equation above, a *UIP* is a collection of terms. The second idea which is presented in Section 3.1.2: expands user query by conflating the closing related terms in a *UIP* with the user query. The expanded query is submitted to a search engine which returns a set of URLs that are presented to the user.

3.1.3 Query Expansion

Query expansion represents the testing phase. In this phase the query terms entered by the user is expanded with the top k terms which were collected in training phase. The top k terms are determined by calculating the contextual similarity of terms in the *UIP* and user query terms. The contextual similarity is calculated using NID (Li, Chen, Li, Ma, and Vitányi, 2004b) as explained in the previous section. The weight w_i is used for identifying the most relevant user interests and its application is described below. After extracting the contextually similar and closest term to user query, we divide the weight w_i of each term a_i in profile p with the exponential over difference of current date and date of activity as shown in equation 3.5. The date of activity is used to maintain the validity of profile

$$P_{temp} = p$$

$$P_{perm} = \frac{p}{e^{c(t)-a(t)}} \quad (3.5)$$

where $c(t)$ is current date and $a(t)$ is date of activity. The division operation reduces the importance of a profile as it gets older. Thus, it takes care of changing user interest. Note that, if a profile consists of some terms that got updated recently, their weight increases and also their date of activity changes to the most recent one. In other words, a collection of terms which

3.2 Exclusively Yours' Algorithm

got introduced long time back and has not been updated lately, means it no longer reflects user interest. The final profile P_{final} can be calculated as shown in equation (15), which is a union of P_{temp} and P_{perm} .

$$P_{final} = P_{temp} \cup P_{perm} \quad (3.6)$$

The expanded query is submitted to a search engine of user choice. We decided to choose the value of k as four i.e. conflate the top four or less contextually similar terms with the user query. (Phelps and Wilensky, 2000) reported in their research that five terms are sufficient to determine web resource uniquely.

3.2 Exclusively Yours' Algorithm

In this section, we briefly explain about the web search APIs used and give an overview of the algorithm behind the proposed approach. Figure 3.3 presents a snapshot of Exclusively Yours' user interface. All the three web search APIs provide the same type of functionality. We can use web search APIs to request query, receive total number of results, URLs, snippets, and title. Although the APIs are provided for free, they impose certain restrictions like the number of query terms, the number of queries that can be issued in one day, and the number of results in one set. Google and Yahoo return 10 results in one set, whereas Naver returns all the results as an xml file. We developed Exclusively Yours' using Java technologies, HTML Tidy, DOM API¹, and Apache

¹<http://tidy.sourceforge.net>

3.2 Exclusively Yours' Algorithm

Server. The user is expected to login and choose a particular search engine before requesting a query. Individual user information such as query submitted, results returned (snippets and titles), total number of results, and web pages clicked by the user are logged in the database which is used later for experiments. Using web search APIs has many advantages: The system is dynamic, personalization is based on data readily available to the search engine, and we don't need to invade user personal information. Following is a brief description of Exclusively Users' algorithm. The algorithm itself doesn't deserve

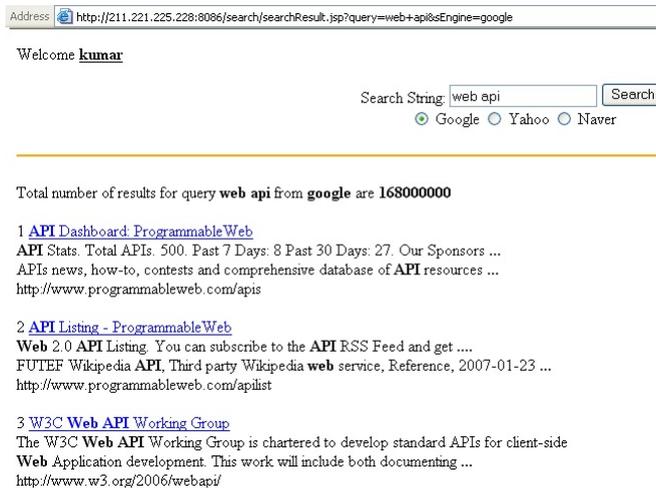


Figure 3.3: A Snapshot of Exclusively Yours' user interface

much explanation as it has already been explained in previous sections. In brief, procedure PERSONALIZE-RESULT forms the core part of Exclusively Yours' system. It primarily does two jobs: (1) creates a profile using procedure CREATE-PROFILE, (2) extracts anchor text along with its surrounding text

3.2 Exclusively Yours' Algorithm

using EXTRACT-ANCHOR Procedure. Moreover, it conflates user profile with the query terms and observes user browsing behaviour, i.e. URLs clicked by the user. The procedure CREATE-PROFILE creates user profile using terms, their weights and activity date stored in index file. The procedure EXTRACT-ANCHOR saves terms along with their weights and activity date extracted from the hub URLs of clicked web pages. Figure 3.4 presents a snippet of code that receives and presents the user with URLs, snippet, title, and the total number of results for user query qs. The first if condition investigates, if user selected yahoo search engine, in that case, it creates an instance of YahooBean. The method setDirectiveArg() sends the user query to the yahoo server. The first 10 URLs, snippets and title are returned using the method, getResult(), getVectorSnippet(), and getVectorTitle() respectively. The method getTotalNumberOfResults() returns the total number of results returned by the search engine. Finally, the for-loop presents URL, snippet, and title to the user. The displayed 10 results are logged in the database. On clicking any of the URL, displayURL.jsp executes, which updates the record pertaining to URL clicked and redirects the browser to the appropriate URL. Further, anchor text extraction module executes to extract the anchor text and its surrounding text from the hub pages of clicked URL as shown in Figure 3.4. We developed a class HTMLParser that takes two inputs, the hub URL and the URL of clicked page. This code starts with the creation of an object of type HTMLParser. Finally the method extractAnchorText() of HTMLParser uses HTML Tidy to fix mistakes if any in the hub URL. After fixing the hub URL, it uses DOM

3.3 Experiments

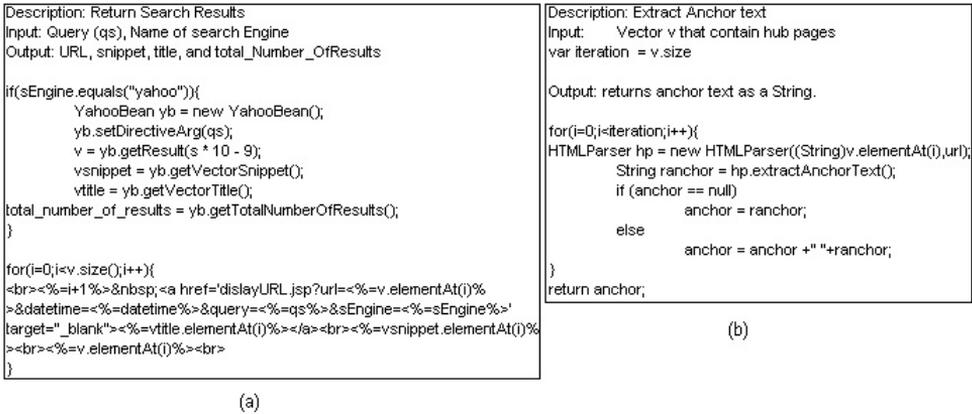


Figure 3.4: (a) Display URLs, snippet and title (b) extracts anchor text and its surrounding text from hub URLs.

API to extract anchor text and surrounding text from the hyperlink that links to URL clicked by the user.

3.3 Experiments

The objective of query expansion is to improve the precision of returned web search results. Hence, we evaluate our system over a large set of queries. We use two measurements to compare the performance of Exclusively Yours' personalized web search system with the original search engine: Average Rank and Discounted Cumulative Gain. To demonstrate the effectiveness of Exclusively Yours', Section 3.3.1 presents the parameter and data sets used for experiment. The metrics used for evaluation are described in Section 3.3.2.

Section 3.3.3 reports the comparison of Exclusively Yours' with some of the closely related personalized approaches, and section 3.3.4 compares Exclusives Yours with non-personalized search engines.

3.3.1 DataSet

In this section, we demonstrate the status of our system as it passes through various phases and the output thereafter. For the experiment purpose, our system was used by 15 volunteers over a period of one month. The 15 volunteers were students, professors, and researchers from various departments at Inha University and Suwon University in Korea. To test the full capability of our system we deliberately selected three volunteers from different departments such as Computer Science, Metallurgy, Biology, History, and Chemistry. In the span of one month, we collected approximately 2450 queries. The first fifteen days correspond to training phase, and the rest of fifteen days correspond to testing phase. Our system learns user behaviour and construct index file of extracted terms in the first 15 days. For the rest of 15 days, it does both the things; construct index file, update user profile and return personalized results. The number of days selected for training phase is purely empirical. We observed that a user needs at least 50-65 queries over a period of one week such that the proposed system can infer his/her interests. Just to make sure that a user inputs 50-65 queries, we assigned a period of 15 days for training. Apart from that, if a user thinks that he is searching something which is unconventional and should not be observed, he can choose to switch off the personalized system

and use the search results from the original search engine. In that case, our system neither extract terms nor does query expansion. Finally, we have tested all our results for test of significance (t-Test). The test condition is whether the personalized search result set improves the search quality when compared with the search result set of non-personalized search engine.

3.3.2 Evaluation Metrics

The metric Average Rank Manning et al. (2008) is used for measuring the quality of personalized search. The Average Rank (AR) of a query q is defined as shown in equation 3.7.

$$AR_q = \frac{1}{|V|} \sum_{p \in V} R(p) \quad (3.7)$$

where $R(p)$ is the rank of URL p . The final AR over all the queries for a use is computed as shown in equation 3.8. Smaller value of AR indicates better placement of results.

$$AR = \frac{1}{|Q|} \sum_{q \in Q} AR_q \quad (3.8)$$

Second metric that we used for measuring the quality of results is Cumulative Gain Järvelin and Kekäläinen (2002). A higher value of Gain Vector symbolizes more relevant results and vice versa. For example: if the highest value of CG is 20 in scenario1 and 12 in scenario2, that implies scenario1 has more highly relevant or relevant results as compared to scenario 2. The Cumulative Gain

Vector is calculated as shown in equation 3.9.

$$CG = \begin{cases} G(1) & \text{if } i = 1 \\ CG(i-1) + G(i) & \text{otherwise} \end{cases} \quad (3.9)$$

Third metric used for measuring the ranking quality is Discounted Cumulative Gain (*DCG*). Järvelin and Kekäläinen (2002). *DCG* is particularly useful when results are evaluated at difference relevance levels (highly relevant, relevant, and not relevant) by assigning them difference gain values. The idea behind *DCG* is, the greater the rank, the less accessible that URL is and hence less valuable it is to the user. The equation 3.10 shows the formulae used for computation of *DCG*.

$$DCG = \begin{cases} G(1) & \text{if } i = 1 \\ DCG(i-1) + \frac{G(i)}{\log_b(i)} & \text{otherwise} \end{cases} \quad (3.10)$$

For the purpose of this experiment, we have used three different relevance level $G(i)=2$ for highly relevant results and $G(i)=1$ for relevant results and $G(i)=0$ for not relevant results. Also b is the parameter of penalization; we have taken value 2 for b .

3.3.3 User Profile Efficacy

The objective of this section is to demonstrate the effectiveness of our proposed personalization method when compared with similar personalized search methods. We constructed *UIPs* using different methods: anchor text and its

3.3 Experiments

surrounded text (referred as anchor text), title, meta-tag keywords, and user browsing history. Note that, a user browsing history is available through the browser cache or using JavaScript's history object. To construct a *UIP* using title, we extracted title from the clicked URLs. To construct a *UIP* using meta-tag keywords, we extracted meta-tags from the clicked URLs. We were able to extract approximately 1050 browsed URLs from the browse cache. The P@10 was used as a performance measure which is shown in 3.5 which depicts that both anchor text with its surrounding terms and browser cache have almost same performance whereas user profile constructed using title of web page gives least performance. The reason for approximately similar performance of user profile constructed from anchor text along with its surrounding text and browser cache can be because both of them primarily represent extraction of anchor text from URLs. The difference lies primarily with the source of URLs. In the first case, i.e. anchor text user-based profile, the anchor text along with its surrounding text is extracted from the clicked URLs. Whereas in the second case, i.e. browser cache user-based profile, the anchor text along with its surrounding text is extracted from all the URLs that have been accessed by the user and are stored in cache. The browser cache based user profile can be thought of as it encompasses the URLs that were clicked by the user added with other URLs browsed by the user. Note that there is a slight drop in the performance of browser cache based user profile. It is because of some noise in user profile that gets induced due to URLs that were not clicked by the user but typed in the browser and browsed for some general information. Or

3.3 Experiments

they may be some pop-ups. The lower performance of title-based user profile and meta-tag keywords can be explained by the way a developer develop web pages. Web developers deliberately scribble such types of title and meta-tag keywords that are misleading and are not related to the content of that web page. Since, the efficacy of user profile using our method is significantly better than the other similar methods and is quite close to user profile constructed using browser-cache, we refrained going on with further experiments i.e. *DCG* and *NDCG*.

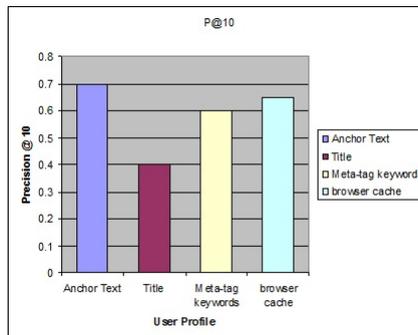


Figure 3.5: Efficacy of *UIP* constructed using different methods

3.3.4 Personalized vs. Non-Personalized Results

We shall now evaluate how the rankings of a non-personalized search engine and a personalized search engine differ based on the valuation we collected from our volunteers. We report two types of results here: one shows the results for an individual user and the other for a group. We found that, the personalized

search engine returned more relevant results as compared to results returned from a non-personalized search engine. However, the same query when issued by multiple users, received differed result sets and also the user's rating was better. Figure 3.6 presents the CG curve for rank 1-30; the plotted graph com-

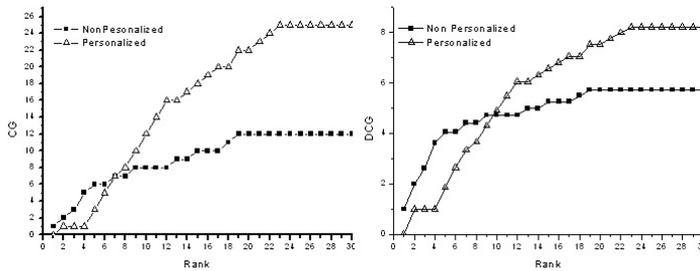


Figure 3.6: (a) Cumulative Gain (CG) Curve for an individual user query (b) Discounted Cumulative Gain (DCG) for an individual user query.

pares the user's evaluation between non-personalized search engine results and personalized search engine results. Note that, the CG of a non-personalized search engine is flat at some places which indicates non relevant results. The steeper the curve, the more highly relevant results and the flat curve indicates not relevant results. The CG curve of a non-personalized search engine trails a horizontal line at rank 19 and onwards. This means, all of the relevant documents were available until rank 19. On the other hand, personalized search engine rank goes horizontal after rank 25. Moreover, the personalized search engine plot is steeper as compared to non-personalized search engine plot. Another metric that is worth noticing is the value of CG . The highest value of CG

for non-personalized search engine is 19, whereas for the personalized search engine, it is 25. The higher value for personalized search engine shows that more relevant results were presented to the user at higher ranks.

Figure 3.6(b) shows the *DCG* curves for ranks 1-30, that compares a non-personalized search engine results with the personalized search engine results. The \log_2 of the document rank is used as the discounting factor for the computation of *DCG*. One important thing to notice in this plot is the *DCG* of first 10 results. The *DCG* of initial results for personalized search engine is a little bit lower than the original non-personalized search engine results. There were a few such cases that this kind of situation occurred. However, from the plot for average *DCG* as shown in Figure 3.7(a) and the plot for average *CG* as shown in Figure 3.7(b), it is evident that results returned by personalized search engine have higher *DCG* thus representing better result quality. We

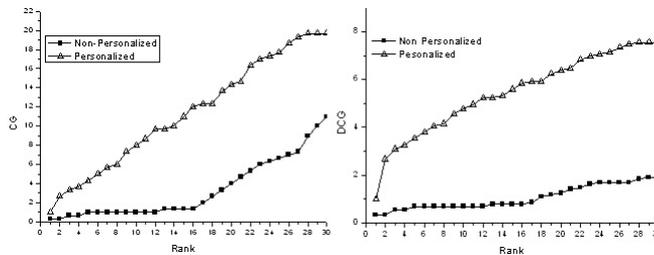


Figure 3.7: Average Discounted Cumulative Gain (*DCG*) Curve and (b) Average Cumulative Gain (*CG*)

investigated the reason for such discrepancy. The explanation follows. Our results are based on user interest and not based on query intent. We have been

3.3 Experiments

able to derive user interest and expanded the same with the user query but still there is a need to derive the intent behind the query. It will be an interesting future work to learn how to derive query intent and what effect does it have on search quality. Figure 3.8(a) shows the Average Rank (AR) for 5 departments

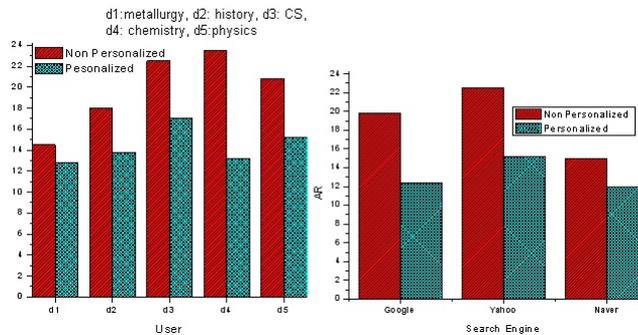


Figure 3.8: (a) Average Rank vs. each department (b) Average Rank vs. Search Engine

(metallurgy, history, computer science, chemistry, physics). It is clearly apparent that overall improvement is 37.6%. The best results are obtained for the chemistry department with improvement of 43.7%. We also learnt through our experiments that the need of personalization varies from query to query. As a matter of fact in some queries personalization produced bad results. For example, a user requested the query rank aggregation: a non-personalized search engine returned highly relevant results. The same query when conflated with his interests resulted in an increase in AR, which means bad quality results. We observed that in 5% of all the queries, the results returned lead to increase in AR. In another case, a user from physics department requested the query CNT

(Carbon Nano Tube), the top 17 results returned by a non-personalized search engine were all irrelevant, and hence in that case, there was significant improvement with personalization. This is another pointer where some improvement is required. We argue that if a system can distinguish between queries that require personalization and those that don't, then one can choose when and when not to apply personalization. This is an interesting future work, which we wish to carry on. Finally, Figure 3.8(b) shows the AR improvement when our personalized system is compared with non-personalized search engine. Over the entire experiment, our personalized system improved our Google, Yahoo, and Naver by 37.6%, 32.4%, and 20% (significant level with $p < 0.05$), respectively.

3.4 Conclusions

In this chapter, we proposed a personalized search method, Exclusively Yours', that infers user interests from user click through behavior. The URLs that a user clicks or downloads is used for the construction of *UIP*. Further, we extract the anchor text and its surrounding text from the associated hub pages of the URLs clicked by the user. In order to use extracted terms later for query expansion, we quantify the importance of each term by assigning a weight. We evaluated our personalized system with Google, Yahoo, and Naver using Cumulative Gain (*CG*), Discounted Cumulative Gain (*DCG*), and Average Rank (*AR*). We found that the proposed approach had significant improvement over non-personalized search engine except for 5% of the queries where personaliza-

tion had a negative impact. The average AR improvement is reported to be 30%.

We also observed that a *UIP* built from anchor text generates a better quality of search results resulting in user satisfaction, nonetheless, it has its own limitations. Anchor text was also found to contain some noise in the form of terms, such as 'next', 'go to', 'click here', etc. However, these anchor text was added without any maligned intention, unlike meta-tag keywords that contained terms not related to the Web document contents and were deliberately added to increasing the ranking of Web document. To further improve the quality of *UIP*, we propose a method that constructs a *UIP* from the tag annotations to the user clicked documents. Tags are annotated to a document by a wide variety of users, it is non-maligned, has rich content, and therefore we believe that it will result in a more enriched *UIP*. Personalization search methods that use tag annotations from a folksonomy system are termed as folksonomy based personalized search method.

4

Matrix factorization for building Clustered User Interest Profile: A folksonomy based personalized search

Quick ways to summarize documents, low latency to access documents, and convenient mechanisms to sharing them are all part and parcel of our daily lives. There is indeed a very large number of documents to deal with¹. Naturally, everyone will benefit if there exist smart programs to manage document

¹<http://googleblog.blogspot.in/2008/07/we-knew-web-was-big.html>

collection, tag them automatically, and make them searchable by keywords. To satisfy such needs, the multimedia, information retrieval, and computer vision communities have, time and again, attempted automatic document annotation, as we have witnessed in the recent past (Uren, Cimiano, Iria, Handschuh, Vargas-Vera, Motta, and Ciravegna, 2006). While many interesting ideas have emerged, not much attention has been paid to the direct use of automatic annotation for document search. Usually, it is assumed that good annotation implies quality document search.

One way of annotation that was widely discussed in the research community is the Social Semantic Web. It largely depends on pre-conceived ontology. However, due to a large amount of initial efforts demanded from web developer community, it did not achieve its success as was expected unlike Web documents which were/are hugely successful in realizing the current Web. Second impediment is that there is huge learning curve associated with Semantic Web. Unlike HTML where a layman can get started with building an HTML document after a couple of hours. Getting to grips with RDF/XML, SPARQL, and the other core technologies is a big ask for most developers. To then get useful semantic web applications out of these takes a couple more exhausting jumps of complexity, for instance, SWOOGLE - a semantic search engine, has reported that about one-third of the RDF files that it has harvested contains errors (Ding, Finin, Joshi, Pan, Cost, Peng, Reddivari, Doshi, and Sachs, 2004). Social Web has emerged as a hope that stands between the conventional Web and the Social Semantic Web. It stands for the culture of participation and

collaboration on the Web. Structures emerge from social interactions: social tagging enables a community of users to assign freely chosen keywords to Web resources. The structure that evolves from social tagging is called folksonomy and recent research have shown the exploitation of folksonomy structures is beneficial to information access.

In the previous chapter, Chapter 3, I proposed a non-folksonomy based method for personalized search that builds a *UIP* from the model proposed in Figure 1.1. The anchor text was used as a feature that is modelled as a user interest, and it was extracted from the hub pages of the clicked Web documents in the user search history. In this chapter, I propose another feature based approach to user profiling that first builds a *UIP* from the tags annotated to documents clicked by the user. Further, the tags in the *UIP* are grouped together into meaningful clusters, a *CUIP*, as perceived by the user. For ex: if a *UIP* consists of following terms, $[java, programming, travel]$, then based on user inclinations a *CUIP* could be, $[[java, programming], travel]$. For the same *UIP*, another *CUIP* could be $[[java, travel], programming]$. The former *CUIP* represents the context of term *java* as *programming*, whereas, the later *CUIP* represents the context of term *java* as *travel*. To discover hidden semantics, matrix factorization techniques are used in this work. This is to say, the proposed methods in this chapter are also based on feature based user profiling, refer Figure 1.1, where feature is tag annotations to Web documents clicked by the user. A profile is further enriched by discovering hidden semantics in its *UIP*, such profile is called as *CUIP*.

This chapter makes the following contributions:

1. We propose two methods to build a *CUIP* for personalized search: one that uses Singular Value Decomposition (SVD) to generate *svdCUIP*, and the other a variation of SVD, *modSVD*, to generate a *modSvdCUIP*. A set of pairs of the form (t, tw) , where t is a tag and tw is the accumulated weight of the tag t , constitutes a User Interest Profile (*UIP*). A *CUIP* is defined as a set of term clusters, where each term cluster consists of semantically related tags of user interests and tag weights.
2. An automatic evaluation method is proposed to test the proposed methods with the baseline search and folksonomy based personalized search approaches.
3. We performed experiments to evaluate the proposed methods on two different data sets. The first data set, called custom data set, was created from the search histories of 12 volunteers. This data set was organized to establish the ground truth for the evaluation of clustering tendency and clustering accuracy of *CUIPs* generated by the proposed methods. The second data set is a much bigger data set harvested from the AOL search query log. This data set was used to test the improvement in personalized search for the two proposed methods, and their comparisons with other methods.
4. Our results show that personalized search using the *modSvdCUIP* is better than using the *tfUIP* (*term frequency UIP*) (Noll and Meinel, 2007) and

4.1 Aggregating tags from user search history

tfIdfUIP (term frequency Inverse Document Frequency UIP) (Xu, Bao, Fei, Su, and Yu, 2008), and exhibits modestly better performance than the *tfIdfCUIP* (Andriy, Jonathan, Bamshad, and Robin, 2008) and *svdCUIP*. Each cluster, in the cluster structure *CUIP*, identifies a topic, and the application of *CUIP* helps disambiguate the context of user query, which is particularly needed for vague queries.

4.1 Aggregating tags from user search history

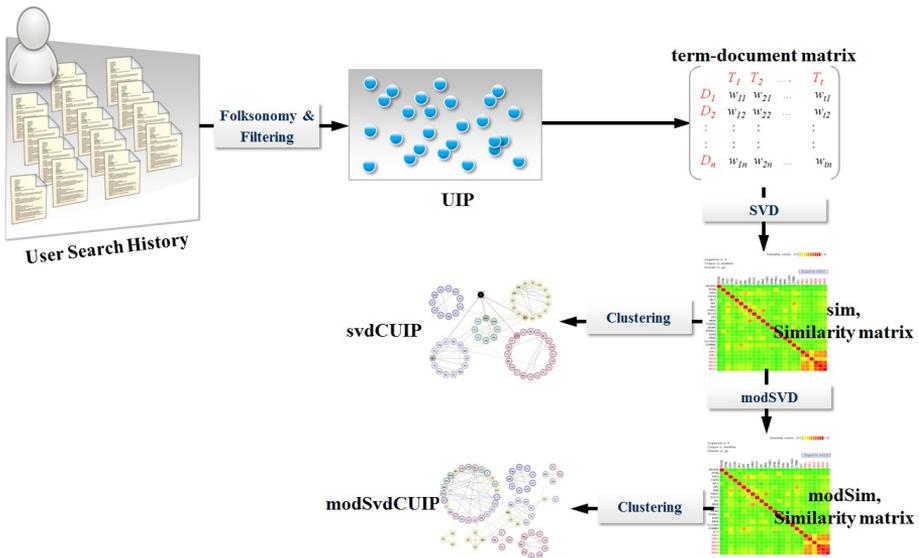


Figure 4.1: System Architecture of CUIP based Personalized Search

Figure 4.1 presents the overall architecture of CUIP based personalized

4.1 Aggregating tags from user search history

Table 4.1: Clicked Web documents and tags attached to the documents

URL	Tag
d_1	java, application
d_2	java
d_3	travel
d_4	iphone, game
d_5	iphone, application

search. When a user clicks on a Web document, it indicates the user interest in that document (Agichtein, Brill, and Dumais, 2006). A user search history provides a collection of the Web documents clicked by the user. Let's call the collection set U . For each Web document $u \in U$, its annotations (tags) are extracted from a social bookmarking service. The tags are stemmed during extraction. Let T be a set of stemmed tags extracted from the social bookmarking service. Note that it is not necessary for the user to have previously used these tags for annotation. The extracted tags were annotated to the documents by the users of the social bookmarking service. Let R be a binary relation between U and T . In order to express that a Web document $u \in U$ is in a relationship with a tag $t \in T$, we write $(t, u) \in I$, which can be read as "the tag t is a topic of the Web document u ". A user context in Table 4.2 is derivable from the relations between Web documents and the tags in Table 4.1. In Table 4.2, each row has a tag in its first column, followed by tag-values, each denoting the importance of the tag for the document clicked by the user. The higher the value, the more useful the tag is for describing the document. Each tag,

4.1 Aggregating tags from user search history

Table 4.2: A user context derivable from Table 4.1

	d_1	d_2	d_3	d_4	d_5
iphone	0	0	0	1	1
java	1	1	0	0	0
game	0	0	0	1	0
travel	0	0	1	0	0
application	1	0	0	0	1

t , annotated to a Web document, d_i , has a tag-value $w(t, d_i)$ representing the number of times d_i has been annotated with t . For example, $w(java, d) = 1$ means the tag *java* has been used to annotate the document d once. A tag weight, $w(t)$, is an aggregated value of t originating from the resource profiles (*RPs*) of multiple documents. It is very likely that the same tag may originate from multiple documents, each with a potentially different tag-value for the tag. We use the standard result set fusion technique, shown in Equation 4.1, to aggregate the tag weight, $w(t)$, from the Web document collection $|U|$.

$$w(t) = \sum_{i=1}^{|U|} w(t, d_i) \tag{4.1}$$

A *UIP* is constructed by collecting all the tags along with their tag weights. For example, the *UIP* for the user context in Table 4.2 would be [*java* : 2, *game* : 1, *application* : 2, *travel* : 1, *iPhone* : 2].

Similar to the well-known *term frequency * inverse document frequency* for documents in IR, the same can be modelled in constructing a *UIP*. The $tf*idf$ multiplies the normalized tag frequency $\frac{td[i][j]}{|td[j]|}$ by the relative distinctness of

4.2 Latent Semantics in UIP

Latent semantics connotes hidden relationships among terms that may exist, but are not explicitly visible. The latent semantics between terms can be discovered by observing the patterns between them such as co-occurrence. Extracting latent semantics between terms helps improve the usefulness of the *UIP*. Co-occurrence between tags can be classified into two types:

1. Two or more tags that annotate the same document: there exist first-order co-occurrences between the tags.
2. Two or more tags that do not annotate the same document; however, there is some hidden relationship between them because they may be related to similar topics: there exist second-order co-occurrences between the tags.

We propose a system that discovers semantically related tags and groups them together, even though they are not identical or do not annotate the same document. The approaches to establishing latent structures in a *UIP* are based on the assumption that the more similar tags are, the more closely related they are.

4.2.1 Computing the tag-tag Similarity matrix

Co-occurrence similarity derives similarity between two or more tags that annotate the same document. The degree of similarity is calculated using the

co-occurrence frequency, called first-order co-occurrence similarity. Another type of co-occurrence similarity is second-order co-occurrence similarity that derives similarity between two tags that do not annotate the same document, but both are related to at least one other tag that annotates the document. It is analogous to finding a friend of a friend and quantifies the degree of friendship relationship. A straightforward approach to measuring the similarity between two tags is to use the Jaccard coefficient between their tag vectors. An alternative approach is to employ matrix factorization on the tfIdf matrix.

We use two matrix-factorization-based methods to calculate the tag-tag similarity matrices. Latent Semantic Analysis (LSA) (Scott, Susan, George, Thomas, and Richard, 1990) uses a matrix factorization technique, Singular Value Decomposition (SVD), to create a new abstract representation of a document corpus in the latent squares sense. The SVD decomposes the tfIdf matrix into three matrices, $A = USV^T$: U , a tag by dimension matrix; S , a diagonal matrix of singular values; and V , a document by dimension matrix. The SVD translates the tag and document vectors into a space determined by the rank r of matrix A . The first r columns of matrix U and matrix V form an orthogonal basis for the tag by document matrix's tag space and document space, respectively

One advantage of the SVD is that it is possible to find a low-rank approximation of the original matrix that removes noise. When we select the k largest singular values from S and their corresponding singular vectors from U and V , we get the rank k approximation, $A_k = U_k S_k V_k^T$, where k is the dimension

reduction parameter. The left singular vectors provide a mapping from the tag space to a newly generated abstract space, while the right singular vectors provide a mapping from the document space to a newly generated space. To compute the tag-tag similarity matrix, we compute U_k , a low-rank approximation of U matrix. After the dimensionality reduction step, the term-term similarity matrix, Sim_k , is computed by using Equation 4.4.

$$Sim_k = U_k S_k (U_k S_k)^T = U_k S_k S_k^T U_k^T = U_k S_k^2 U_k^T \quad (4.4)$$

Dimensionality reduction reduces noise in the tag-tag similarity matrix, resulting in richer relationships between tags that reveals the hidden semantics present in the document corpus. The value of Sim_{ij} in Sim_k represents the similarity between tags t_i and t_j . The higher the value, the higher the relatedness is between the tags. In theory, the value of Sim_{ij} captures both orders of co-occurrence similarities between t_i and t_j across the corpus. That is, the value is based on the transitive relation between terms due to a chain of intermediate terms that link the terms t_i and t_j . Note that it is not necessary for t_i and t_j to belong to the same document, but there should be a chain of terms that link them. Two factors influence the magnitude of similarity value Sim_{ij} : 1) the number of intermediate tags, or the length of the chain that connects t_i and t_j ; and 2) the tag-weights of the intermediate tags. The example below shows the step-by-step procedures to obtain the similarity matrix, Sim_2 , by applying Equation 4.4 on the tfIdf matrix, A .

Note that there exists a disparity in the similarity values in Sim_2 . The reason is that the user context in Table 4.2 indicates that the tag "iphone" is co-located with the tags "game" and "application", and not with the tag "java". The SVD process has successfully captured the relationships "iphone" and "game", and "iphone" and "application", which is a first-order co-occurrence relationship. Also, it has successfully discovered the hidden relationship between "iphone" and "java", because of the intermediate tag "application" that co-occurs with "java" and "iphone". However, the magnitude of relationship is misleading: it suggests a stronger relationship between "java" and "iphone" (0.3517) compared to "iphone" and "application" (0.1235), and "iphone" and "game" (0.0481).

4.2 Latent Semantics in UIP

$$\begin{aligned}
 A &= \begin{matrix} & d_1 & d_2 & d_3 & d_4 & d_5 \\ \begin{matrix} iphone \\ java \\ game \\ travel \\ application \end{matrix} & \begin{pmatrix} 0 & 0 & 0 & 0.661 & 0.661 \\ 0.661 & 1.3219 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1.162 & 0 \\ 0 & 0 & 2.32 & 0 & 0 \\ 0.661 & 0 & 0 & 0 & 0.661 \end{pmatrix} \end{matrix} \\
 U &= \begin{pmatrix} 0.00 & -0.16 & -0.59 & 0.28 & -0.74 \\ 0.00 & -0.92 & 0.26 & -0.27 & -0.1 \\ 0.00 & -0.13 & -0.75 & -0.45 & 0.46 \\ 1.00 & 0.00 & 0.00 & -0.00 & 0.00 \\ 0.00 & -0.32 & -0.14 & 0.00 & 0.48 \end{pmatrix} \quad U_2 = \begin{pmatrix} 0.00 & -0.16 \\ 0.00 & -0.92 \\ 0.00 & -0.13 \\ 1.00 & 0.00 \\ 0.00 & -0.32 \end{pmatrix} \\
 S &= \begin{pmatrix} 2.32 & 0.00 & 0.00 & 0.00 & 0.00 \\ 0.00 & 1.53 & 0.00 & 0.00 & 0.00 \\ 0.00 & 0.00 & 1.4 & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.94 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.00 & 0.33 \end{pmatrix} \quad S_2 = \begin{pmatrix} 2.32 & 0.00 \\ 0.00 & 1.53 \\ 0.00 & 0.00 \\ 0.00 & 0.00 \\ 0.00 & 0.00 \end{pmatrix} \\
 Sim_2 &= U_2 S_2^2 U_2^T
 \end{aligned}$$

$$\begin{aligned}
 & \begin{matrix} & iphone & java & game & travel & application \\ \begin{matrix} iphone \\ java \\ game \\ travel \\ application \end{matrix} & \begin{pmatrix} 0.0621 & \mathbf{0.3517} & 0.0481 & 0.00 & \mathbf{0.1235} \\ 0.3517 & 1.9928 & 0.2726 & 0.00 & 0.6996 \\ 0.0481 & 0.2726 & 0.0373 & 0.00 & 0.0957 \\ 0.00 & 0.00 & 0.00 & 5.3914 & 0.00 \\ 0.1235 & 0.6996 & 0.0957 & 0.00 & 0.2456 \end{pmatrix} \end{matrix} \\
 &= \begin{matrix} & iphone & java & game & travel & application \\ \begin{matrix} iphone \\ java \\ game \\ travel \\ application \end{matrix} & \begin{pmatrix} 0.0621 & \mathbf{0.3517} & 0.0481 & 0.00 & \mathbf{0.1235} \\ 0.3517 & 1.9928 & 0.2726 & 0.00 & 0.6996 \\ 0.0481 & 0.2726 & 0.0373 & 0.00 & 0.0957 \\ 0.00 & 0.00 & 0.00 & 5.3914 & 0.00 \\ 0.1235 & 0.6996 & 0.0957 & 0.00 & 0.2456 \end{pmatrix} \end{matrix}
 \end{aligned}$$

could be beneficial for a clustering algorithm to generate accurate clusters.

$$Sim_3 = \begin{matrix} & \begin{matrix} iphone & java & game & travel & application \end{matrix} \\ \begin{matrix} iphone \\ java \\ game \\ travel \\ application \end{matrix} & \left(\begin{matrix} 0.3577 & \mathbf{0.0275} & \mathbf{0.4404} & 0.00 & \mathbf{0.1349} \\ 0.0275 & 1.0185 & -0.0491 & 0.00 & 0.3035 \\ 0.4404 & -0.0491 & 0.5488 & 0.00 & 0.1422 \\ 0.00 & 0.00 & 0.00 & 2.5903 & 0.00 \\ 0.1349 & 0.3035 & 0.1422 & 0.00 & 0.1354 \end{matrix} \right) \end{matrix}$$

However, even with $k = 3$, the magnitudes of relationship, expressed in similarity values, are rather low for second-order co-occurrence similarity ("iphone" and "java") compared to the first-order co-occurrence similarity ("iphone" and "game" or "iphone" and "application"). This seriously jeopardizes the effectiveness of the clustering algorithm to generate clearly separated clusters. In real scenarios, sparseness of a similarity matrix, Sim_k , could be as high as 90%, which seriously affects the ability of the SVD to correctly discover the second-order co-occurrences. We show in the experiment section the effect of sparseness of Sim matrices on clustering tendency and clustering accuracy.

The second-order co-occurrence similarity values are too small to be detected by clustering algorithms. The experiment results show that the numbers of values in the term-term similarity matrix, greater than 0.5, is small, nullifying the usefulness of SVD to discover 2^{nd} order co-occurrence between terms.

Higher values of $modSim_{ij}$ signify a greater overlap between the two vectors across n dimensions. Thus, it aids in demarcating clusters boundaries, resulting in fine clusters, and also helps induce sense from contextual similarity.

4.2.2 Tag Clustering to generate *svdCUIP* and *modSvdCUIP*

(Scott, Susan, George, Thomas, and Richard, 1990) urged the necessity of clustering in Information Retrieval (IR) tasks. The authors state that IR systems treat each term as independent from others. Treating a term independently may lose the latent contextual information that can make substantial difference in information retrieval tasks. This has motivated us to use clustering in our work.

Term Clustering algorithms generally consist of two phases. The first phase requires computing a term-term similarity matrix, and the second phase uses the matrix to generate clusters of coherent terms. Two major types of clustering algorithms are available: partitioning and hierarchical. The partitioning clustering generates topic clusters, whereas the hierarchical clustering generates cluster hierarchies. Topic clusters are created by grouping similar and closely related terms together into a unified topic. In a cluster hierarchy, terms are placed in the leaves at the bottom of the hierarchy with more specialized topics immediately above them, and so on. Hierarchies are very large and complex in nature. We want hierarchies but not too specific terms. We are, on the other hand, interested in crisp clusters. Therefore, we adapted a hybrid approach that generates a hierarchy, which is further dissected to generate crisp

term clusters. We used the Hierarchical Agglomerative Clustering Algorithm (HAC)(Gower and Ross, 1969) because it fits best when the number of clusters is unknown beforehand. We use distinctness parameter, d , to cut the single hierarchy of clusters to obtain a number of clusters. For instance, Table 4.3 shows the clusters, in the cluster structures *svdCUIP* and *modSvdCUIP*, obtained by applying HAC on *sim₃* and *modSim₃* matrices. The *svdCUIP* has four clusters, and it fails to identify that "iphone" and "game" should belong to the same cluster, whereas the *modSvdCUIP* identifies all the term clusters accurately. It is very important to choose the right value of d to generate appropriate term clusters matching the user's perspective, thus achieving a high clustering accuracy. Figure 4.2 shows a dendrogram output when the similarity matrix *modSim* is input to the HAC. With $d \geq 1.4$, one cluster is created, a hierarchy of all the terms; with $d = 0.4$, there are three clusters; and, with $d < 0.3$, there is a flat list of terms.

At the outset, HAC treats each term as a singleton cluster and then successively merges pairs of clusters until all the clusters have been merged into a single cluster that contains all the terms. Cluster proximity is used to merge clusters. There are three well known proximity measures: single linkage, complete linkage, and average linkage. The single linkage proximity measure is the distance between the closest two points that are in two different clusters, i.e., the maximum similarity between two terms. On the contrary, the complete linkage takes the distance between the farthest two points in two different clusters as the cluster proximity. The average linkage defines cluster proximity as

Table 4.3: Clusters obtained by applying HAC on similarity matrices Sim_3 and $modSim_3$ for $k=3$ and $d=0.35$

Method	Cluster Structure
<i>svdCUIP</i>	[[iphone], [java, application],[game],[travel]]
<i>modSvdCUIP</i>	[[java, application],[iphone,game],[travel]]

the average pairwise proximity, an average length of edges of all the terms from two different clusters. We carried out experiments using the three proximity measures, but this research reports on only the average linkage in the experiment section because it worked better than the others. The explanation in the previous two sub-sections has identified the importance of dimensionality reduction parameter k and distinctness parameter d to generate right number of clusters of good quality. The experiment section shows how to determine the right values of k and d , to generate crisp clusters, without compromising clustering accuracy.

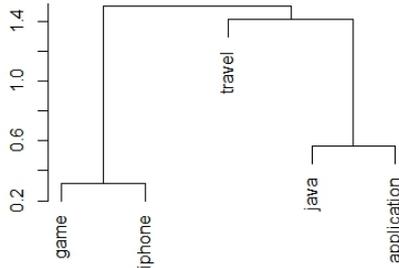


Figure 4.2: Dendrogram visualization for similarity matrix $modSim$

A *CUIP* that results from the application of HAC on a *Sim* matrix obtained

by applying the SVD on a *tfidf* matrix is called SVD based CUIP (*SvdCUIP*). And, a *CUIP* that results from the application of HAC on a *modSim* matrix obtained by calculating the cosine similarity of every pair of tag vectors in the similarity matrix, *Sim*, is called modSVD based CUIP (*modSvdCUIP*).

We also generate a *tfidfCUIP* for each user, an adaptation of (Andriy, Jonathan, Bamshad, and Robin, 2008) approach. A term-term similarity matrix is generated by computing the cosine similarity between tag vectors in the *tfidf* matrix, which is fed to HAC to generate the *tfidfCUIP*. The *tfidfCUIP* is a local cluster structure unlike the (Andriy, Jonathan, Bamshad, and Robin, 2008) approach where the terms in the *UIP* are mapped to a global cluster structure to construct a *CUIP*.

4.3 Personalized Search

This section explains how to use a CUIP for personalized search. The classic SEs compute the relevance between a query and a document using the similarity between the terms that match. They are "One-size-fits-all" in that the search results are the same irrespective of the user. However, a document relevant to a user might not be relevant to another user, though, they both have issued the same query. Thus, the user query as well as its context should be mapped to the term space of the document contents. A query conflated with the contextual terms is called expanded query.

The *CUIP* helps disambiguate a user query by suggesting a matching cluster.

The terms in Web documents and the expanded query are represented as vectors in the space. By using the Vector Space Model (VSM) (Salton, Wong, and Yang, 1975), we compute the similarity between the term spaces of the documents and that of the expanded query to compute the rank of the documents. Let $d = t_1^d, t_2^d, \dots, t_n^d$ be the term vector for a document, where n is the dimension of the term space. Let $qe = t_1, t_2, \dots, t_n$ be the expanded query. The similarity between a document d and a query qe is calculated using Equation 4.6.

$$sim(d, qe) = \frac{d^T \cdot qe}{|d||qe|} \quad (4.6)$$

Given a user query, two steps are executed in the following order: first, find a matching cluster g_m in the user *CUIP* to the query; second, the query and the tags in the matching cluster are fed to the underlying search engine to generate a set of documents that are ranked using equation 4.6.

In this research, we use a class-based Language Modeling (LM) to determine the most closest cluster, for a given query, from the user's cluster structure. This involves computing the similarity between each cluster and query, and choosing the cluster that has the maximum similarity, refer equation 4.3).

$$CUIP = \{g | g = \{t_1, t_2, \dots, t_n\}\}$$

$$P(q, CUIP) = \underset{g \in CUIP}{argmax} \prod (q|g)$$

$$P(q, g) = P(q|t_1, t_2, \dots, t_n)$$

$$P(q|t_1, t_2, \dots, t_n) = \prod_{i=1..n} P(q|t_i)$$

where

$$P(q|t_i) = \frac{\text{count}(q, t_i)}{\text{count}(t_i)} \quad (4.7)$$

4.4 Experimental Evaluation

4.4.1 Data Set and Experiment Methodology

To examine the effectiveness of the proposed methods, we conducted a series of experiments on two different data sets. First, to evaluate the clustering tendency and clustering accuracy of the *CUIP*, we recruited 12 users whose search histories were harvested to construct the first data set, referred as Custom Data Set. Second, to evaluate the quality of personalized search using the proposed methods, we constructed another data set from the AOL search query log¹. For both data sets, the URL-tag annotations were harvested from the Delicious Server using the Delicious API².

4.4.1.1 Custom Data Set and Evaluation Metrics

This data set consists of data from 12 users, mostly master's students, who have considerable experience using search engines. Each user's log of search history for a period of 3 months or 13 weeks was harvested as an RSS feed

¹<http://www.gregsadetsky.com/aol-data/>

²<http://www.delicious.com>

4.4 Experimental Evaluation

from the individual’s Google Search History¹. The RSS feed consists of the following meta data: title of the query input by the user; title of the Web document clicked by the user; the address of the Web document clicked by the user; and, the dates and times at which the queries were submitted. The data set contains 2921 queries, and 6477 clicked Web documents. Of the documents, only 3617 (approximately 55%) were found to be annotated on Delicious.

In clustering, measuring its accuracy and correctness in any certainty is best left to the user’s judgement. Therefore, to establish the ground truth, we asked each user to group related terms extracted from the tag annotations of the Web documents clicked by the user. Each user was asked to manually group related terms together; they were instructed to group terms based on their own understanding rather than the general understanding. The grouping generated manually by the user is called user cluster structure. Generating ground truth manually for evaluation is a normal procedure used in many research works (Bing, 2006, Christopher, Shlomo, and Andrew, 2012, Dom, 2002, Hassan, 2006, Pérez, Zubiaga, Fresno, and Martínez, 2012). Since this process is subjective, we take the average of the scores from all the users as the final score. The whole process was a very labor intensive and time consuming task, which was the primary reason why we opted to experiment with a small set of users.

For each user, two sets of *CUIPs* are generated: one set consists of *svdCUIPs*, and the other of *modSvdCUIPs*. These *CUIPs* are called system generated

¹<http://www.google.com/searchhistory>

4.4 Experimental Evaluation

cluster structures. In each set, a *CUIP* is generated for each combination of dimension reduction parameter k and distinctness parameter d . To construct a *svdCUIP* and a *modSvdCUIP*, the similarity matrices sim_k and $modSim_k$ are generated, respectively. The value for k is initialized to 10, and it increases in an increment of 10 until it reaches 110. This creates 11 sim_k and 11 $modSim_k$ similarity matrices. Similarly, the distinctness parameter d is initialized to 0.03, and it increases in an increment of 0.02 until 0.13, after which it increases in an increment of 0.1 until 0.93 (a total of 14 values). For each user, 154 *svdCUIPs* and an equal number of *modSvdCUIPs* were created. Let the user generated cluster be $C = \{c_1, c_2, \dots, c_n\}$, and the system generated cluster be $D = \{d_1, d_2, \dots, d_m\}$. We chose the Silhouette Coefficient (Rousseeuw, 1987) evaluation metric (unsupervised evaluation) to judge the cluster tendency, and the Fscore (supervised evaluation) evaluation metric to compare the clustering accuracy. The Silhouette Coefficient is a popular method that combines cohesion and separation. Equation 4.8 computes the Silhouette Coefficient for each tag t_i in the system cluster structure.

$$s(i) = \frac{(b_i - a_i)}{\max(a_i, b_i)} \quad (4.8)$$

where b_i is the minimum of all the average distances between term t_i and all the terms in other clusters that do not contain t_i (separation); and, a_i is the average distance between term t_i and all other terms in the same cluster (cohesion). Equation 4.9 computes the average Silhouette Coefficient, \bar{s} , which

4.4 Experimental Evaluation

is the average of the Silhouette Coefficients for all the terms (N) in the cluster structure.

$$\bar{s} = \frac{1}{N} \sum_{i=1}^n s(i) \quad (4.9)$$

An average Silhouette Coefficient is a very useful overall quality measure to measure the clustering tendency of a cluster structure. (Kaufman and Rousseeuw, 1990) provided an interpretation of the average Silhouette Coefficient, \bar{s} , as a measure of evidence in support of a cluster structure: the value of the average Silhouette Coefficient between]0.7, 1.0] suggests strong evidence; between]0.5, 0.7] reasonable evidence; between]0.25, 0.5] weak evidence; and between [-1, 0.25] no evidence.

We also compare the clustering accuracy of the system generated cluster structure with the user generated cluster structure. Fscore(Bing, 2006) measures the extent to which a system generated cluster contains only tags of a particular user generated cluster and all objects of that user generated cluster. Equation 4.10 computes an Fscore by combining precision and recall. Precision, p_i , is the proportion of the tags of user generated cluster c_j in the system generated cluster d_i ; Recall, r_i , is the fraction of matching tags in the system generated cluster d_i that match the tags in the user generated cluster c_j .

$$Fscore_i = \frac{2 * p_i * r_i}{p_i + r_i} \quad (4.10)$$

4.4.1.2 AOL Query Data Set and Evaluation Metrics

The AOL search query log has 20 million Web queries collected from 650,000 users. Each row in the data set contains five attributes: 1) AnonID, an anonymous user id; 2) Query, the query issued by the user; 3) Query Time, the time at which the query was submitted to the AOL search engine; 4) Item Rank, the rank of the Web document clicked by the user; and 5) ClickURL, the address of Web document clicked by the user. We created a dataset of 2000 users, a subset of the total data set. This dataset contains 1,244,714 Web documents, out of which 829,285 documents (approximately 66%) were found to be annotated on the Delicious server. The documents have 212,011 tags annotated to them. Our experiment methodology is geared towards measuring the effectiveness of the proposed personalized search methods and evaluating the improvement they offer in comparison to other methods. Figure 4.3 illustrates the overall evaluation methodology.

4.4.1.3 Experiment set up to estimate the value of k and d

The complete data set is split into two equal parts: the first part is called as the training, or development, data set; and the second part is called as the evaluation data set. The training data set is used to estimate the value of parameters k and d for *svdCUIP* and *modSvdCUIP*, which are directly used in the evaluation dataset to compare the performance of the proposed approaches with the other personalized search approaches. The evaluation data set helps

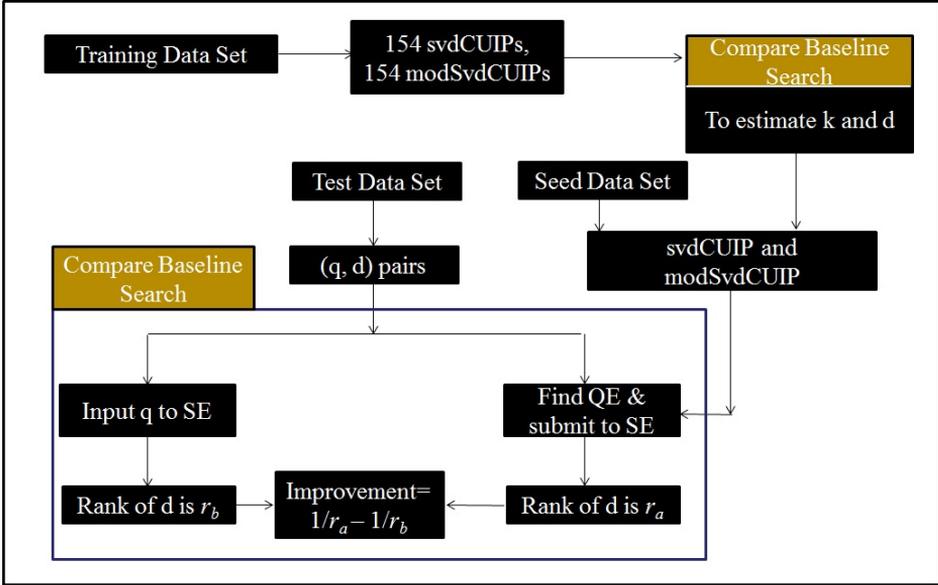


Figure 4.3: Automatic Evaluation Methodology

guard against both under fitting and over fitting.

From the training data set, we construct *UIPs* and *CUIPs*, and pairs of query and associated Web document (referred as target Web document) are extracted from the user search history. For each pair, the query is submitted to the base search engine to calculate the rank of the target Web document, called r_b . Next, the query is expanded with the tags in the matching cluster from the *CUIP*. The expanded query is submitted to the search engine to calculate the new rank of the target Web document, called r_a . The difference in the inverse ranks of the personalized search method and the baseline method is the improvement (Ellen,

1999) of the personalized search method, calculated using equation 4.11.

$$improvement = \frac{1}{r_a} - \frac{1}{r_b} \quad (4.11)$$

The values of k and d , for which the improvement of the proposed methods over baseline search is maximum, are used directly for the further stage of evaluation.

4.4.1.4 Experiment set up to compare the proposed approaches with other approaches

The following steps execute on the evaluation data set:

1. **Indexing:** The contents of each document in the dataset is indexed using Lucene API¹. Lucene is our base search engine, and search using it is referred to in this chapter as baseline search method.
2. **User Profile:** The search history of each user is divided into two parts: the first part, which makes 90% of the entire history, is used for building *UIPs* and *CUIPs*; and the second part, the remaining 10%, is used for generating pairs of queries and URLs, called test collection, to automatically evaluate our methods.
3. **Evaluation:** For each document in the second part, we create a pair that consists of the document itself and the query associated with it. Each

¹lucene.apache.org/core/

4.4 Experimental Evaluation

pair constitutes a test case against which the tasks (a), (b), (c), and (d) below are executed. A test case designates a query and its target Web document.

- (a) For each query and Web document combination in a test case, submit the query to the base search engine to obtain a ranked list of search results. Let the rank of the target Web document in the search result set be r_b . This is the rank of the target document produced by the baseline search method.
- (b) For both *tfUIP* and *tfIdfUIP*, the Web documents in the search result set are re-ranked by calculating the similarity between the *RP* of the Web documents and tags in the *UIP* using equations 2.1 and 2.2, respectively. Let the new ranks of the target document in the re-ranked search result set designated as r_n and r_x for *tfUIP* and *tfIdfUIP*, respectively. Equation 4.11 computes the improvement as the difference between the inverse ranks of the personalized search method and the baseline method.
- (c) Search results are not re-ranked for the *svdCUIP*, *modSvdCUIP*, and *tfIdfCUIP* methods, rather, the query is expanded with the tags in the matching cluster from the *CUIP*. The expanded query is submitted to the search engine to determine a new rank of the target document. The search engine generated the ranking of documents by calculating the similarity between the expanded query and the

document contents using the equation 4.6. The difference in the inverse ranks determined for the personalized search method and the baseline method is the improvement of the personalized search method.

4.4.2 Experiment Results

Sections 4.4.2.1, 4.4.2.3, and 4.4.2.3 determine, for both *svdCUIP* and *modSvdCUIP*, the value(s) of dimensionality reduction parameter k and distinctness parameter d that show(s) strong, or at least reasonable, clustering tendency and clustering accuracy. Section 4.4.2.4 presents an exemplary *modSvdCUIP*. The sections 4.4.2.5 and 4.4.2.6 determine, for both *svdCUIP* and *modSvdCUIP*, the value(s) of dimensionality reduction parameter k and distinctness parameter d using the Improvement as an evaluation metric. And, sections 4.4.2.8 and 4.4.2.9 compare the proposed methods with the other methods using the evaluation metric Improvement.

4.4.2.1 Clustering Tendency

Assessing the presence of clusters in a data set is an important step in cluster analysis. The plot in Figure 4.4 helps visualize clustering tendency in system generated clusters, if any, and also approximates the correct number of clusters in the cluster structure.

It is clear that the cluster structure *modSvdCUIP* has stronger evidence of cluster tendency, whereas the *svdCUIP* shows reasonable or weak evidence of

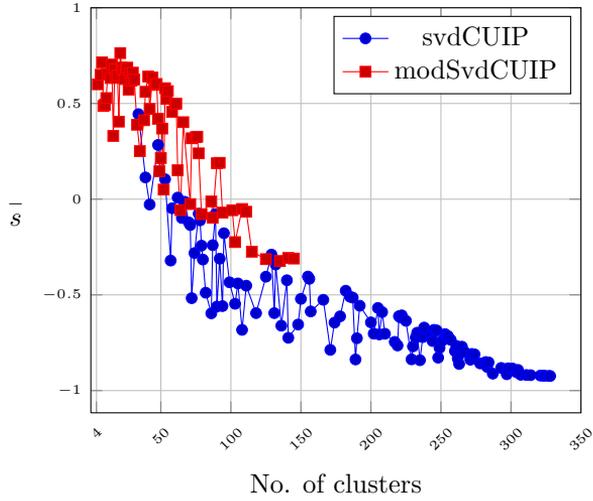


Figure 4.4: Number of Clusters vs. average Silhouette Coefficient plot for *svdCUIP* and *modSvdCUIP*

clustering tendency. We observed that the clustering tendency in a *CUIP* was affected by the ratio of number of zero values to the number of positive values in the tag-tag similarity matrix; the lower the better. The average ratio for the tag-tag similarity matrix *modSim* is 0.9, and 1.68 for the tag-tag similarity matrix *sim*. The maximum and minimum ratios for the *modSim* are 3.2 and 0.6, respectively, and for the *sim*, 6.2 and 1.0, respectively. This evidence explains why the cluster structure, *svdCUIP*, exhibits weak cluster tendency.

Figure 4.4 also indicates that the average Silhouette Coefficient (\bar{s}) decreases as the number of clusters exceeds over 50, which suggests that the best cluster structure was obtained when the number of clusters was around 50. This was

acceptable because the average number of tags in a *UIP* was 594, which could possibly result in 50-70 clusters. However, what is surprising is that, even with less than 10 clusters in the *modSvdCUIP*, the plot shows strong clustering tendency. To try to find the natural number of clusters in a cluster structure, one should look for a knee, a peak, or dip in the plot (Tan, Steinbach, and Kumar, 2005). The plot for the *modSvdCUIP* shows a rise followed by a dip and a peak occurring around when the number of clusters falls between 40 and 60. For the *svdCUIP*, the plot clearly shows a peak when the number of clusters reaches 50.

4.4.2.2 Determining the value for dimension parameter, k , for the Custom Data Set

Figures 4.5 and 4.6 present 3-dimensional plots that show how the average Silhouette Coefficient changes in response to the changes of k and d . The figures help determine the values of k and d for each method. The *svdCUIP* in Figure 4.5 exhibits a clear pattern: for low values of k regardless of d , there is no evidence of clustering tendency; however, for high values of k , between 90 and 100 and low values of d , there is a reasonable evidence of clustering tendency. The weak clustering tendency of the *svdCUIP* is due to the fact that the magnitude of relationship between tags is low. This jeopardizes the ability of clustering algorithms to discern cluster boundaries.

The average Silhouette Coefficient vs. k and d plot in Figure 4.6 for the cluster structure *modSvdUIP* also exhibits a distinct pattern: unlike the *svdCUIP*,

4.4 Experimental Evaluation

the plot for the *modSvdCUIP* shows a strong evidence of clustering tendency for values of $k = 30$ and 40 and middle values of d . It ascertains the fact that increasing the value of d decreases clustering tendency. The *modSvd-CUIP* exhibits a strong clustering tendency because the *modSim* overcomes the limitation of the *Sim* by capturing the information present in second order co-occurrence. Moreover, the information in the *modSim* matrix is less sparse and more robust than the *Sim* matrix.

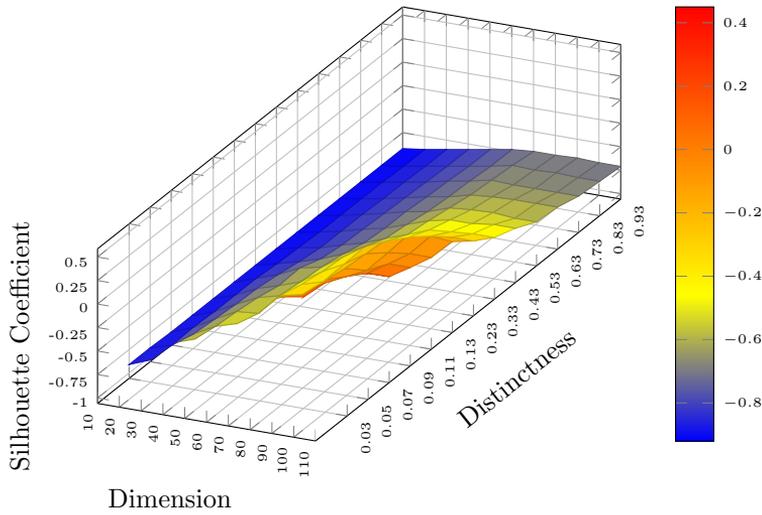


Figure 4.5: A comparison of different value combinations of k and d Vs. average Silhouette Coefficient for *svdCUIP* average linkage

4.4.2.3 Determining the value of distinctness parameter, d , for the Custom data set

The experiment, in this section, focuses on determining the appropriate value of d for the highest accuracy cluster structure. Fscore is used as an evaluation metric to measure and compare the accuracy of the system generated cluster structure with the user generated cluster structure. Figure 4.7 shows the accuracy obtained by each method, and demonstrates that the *modSvdCUIP* has better clustering accuracy than the *svdCUIP*.

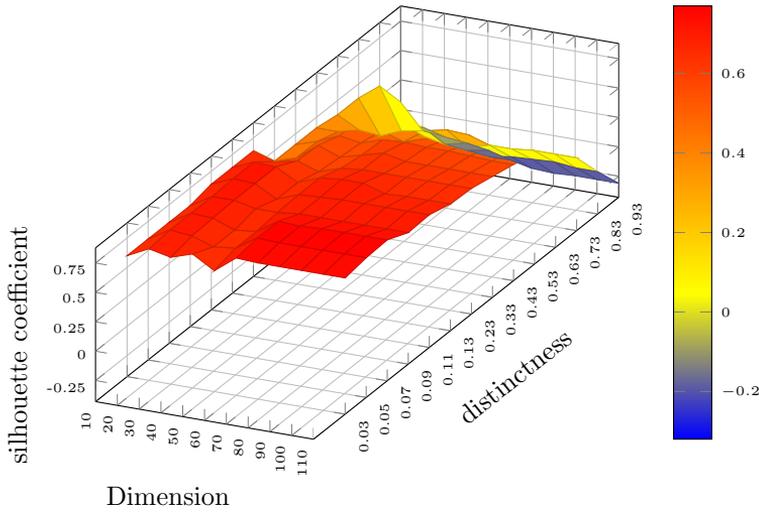


Figure 4.6: A comparison of different value combinations of k and d vs average Silhouette Coefficient for *modSvdCUIP* average linkage

The average clustering accuracy for the *modSvdCUIP* and *svdCUIP* is 0.58

4.4 Experimental Evaluation

and 0.16, respectively; there is a 244% increase in average clustering accuracy. This indicates that the *modSvdCUIP* produced by the *modSvd* is more accurate than the *svdCUIP* produced by the *Svd*. With the *modSvd*, the dimension reduction parameter $k=30$ has higher clustering accuracy than $k=40$. Also, the difference in clustering accuracy between $k=30$ and $k=40$ is marginal. Moreover, both of the curves follow the same pattern, signifying that the clustering accuracies of the *modSvdCUIP* for $k=30$ and $k=40$ are nearly identical with a slightly better performance at $k=30$. The highest clustering accuracy for the *modSvdCUIP* is 0.75, obtained with $k=30$ and $d=0.07$.

Another identical accuracy was exhibited when $k=90$ and $k=100$ in the *Svd*. A careful observation, however, reveals that the *svdCUIP* for $k=100$ shows a marginal improvement over $k=90$, with $d=0.03$ and $d=0.05$. This suggests that either value of the dimension reduction parameter can be used for constructing the *svdCUIP*. The highest clustering accuracy for the *svdCUIP* is 0.55, with $k=100$ and $d=0.03$.

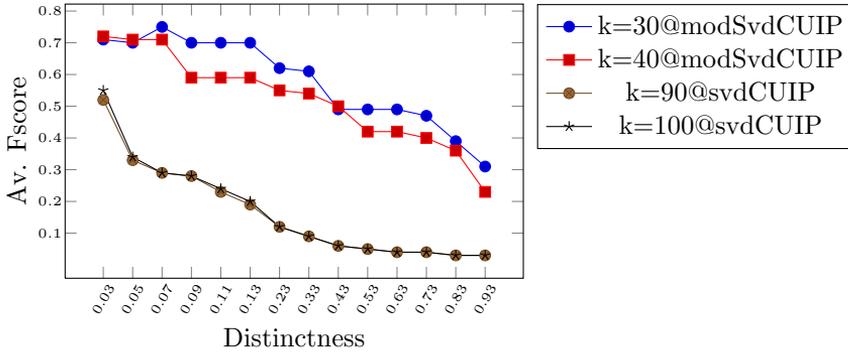


Figure 4.7: A comparison of different value combinations k and d vs *AverageFscores* for the *modSvdCUIP* (when $k=30,40$) and the *svdCUIP* (when $k=90,100$) for average linkage.

These results suggest that the accuracy of the *modSvdCUIP* produced by the *modSvd* is superior to the cluster structure *svdCUIP* produced by the *SVD*.

4.4.2.4 CUIP visualization

We developed our own implementation of Hierarchical Agglomerative Clustering (HAC) in Java. Table 4.4 shows the snapshot of the *modSvdCUIP*, the output of HAC for $d=0.53$, for one of the users. For interested readers, a complete *modSvdCUIP*, *svdCUIP*, and *tfIdfCUIP* is provided in the .3.

The quality of clusters hinges on the level of term coherency, each cluster representing a distinct topic area. Table 4.4 shows a high level of term coherency in clusters, each of which shows user interests such as finance, religion, porn, law, automotive, and entertainment. Moreover, the terms in each cluster are

Table 4.4: Example of cluster structure

Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5	Cluster 6
bank, banking, finance, business, supplier	religion, culture, judaism, jewish, israel	amateur, sex, adult, toys, girls, porn, voyeur	government, patent, trademark, law, legal	auto, automotive, parts, electronics, car	video, movies, film, soccer, game

contextually related, which aids to disambiguate context, synonym terms, and polysemous terms. For instance, Cluster 1 captures the notion of the user’s interests in finance, and disambiguates the context of the polysemous term ”bank”, which in Cluster 1 refers to a financial institution, not to other meanings such as bank as in a river bank.

Cluster 2 indicates that the user is interested in Judaism religion. Synonym terms are clustered together such as ”Jewish” and ”Judaism” in Cluster 2, ”auto” and ”automotive” in Cluster 5, ”movies” and ”film” in Cluster 6. Cluster 5 can be interpreted as that the user is interested in the automotive, in particular cars. She/he might also be interested in the electronic parts of the car. Cluster 6 represents the user’s entertainment options; the user prefers to watch movies or soccer games. The term video is rightly disambiguated by being associated with the term ”movie”.

These results show clear evidence of emergence of topics and contexts that would otherwise be latent in a *UIP*. A *CUIP* is an important source of information that can be effectively used for query suggestion, query classification, Web page recommendation, personalized search, or Web search result ranking.

4.4.2.5 Determining the value of the dimension reduction parameter k for the AOL data set

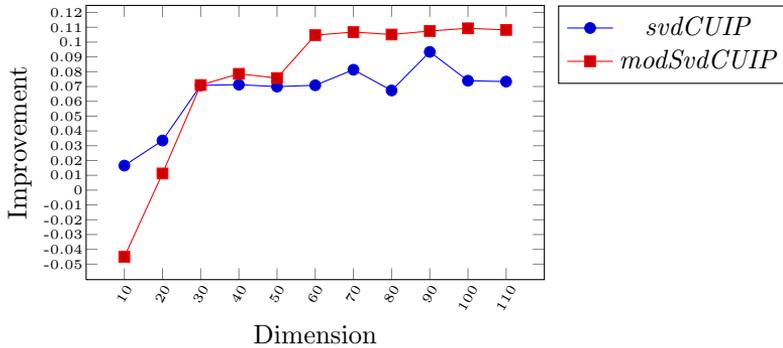


Figure 4.8: Estimating the values of dimension parameter for $svdCUIP$ and $modSvdCUIP$ using the Improvement as an evaluation metric

Since the personalization algorithm relies on the user $CUIP$ to personalize search results, the selection of a proper dimension value is integral to the success of the personalization algorithm. The goal of tuning the dimension parameter is to discover the second order co-occurrence similarity between tags. Figure 4.8 plots the improvement of proposed methods in reference to the baseline search when the value of k changes from 10 to 110 in an increment of 10. It indicates that the $modSvdCUIP$ based personalized search shows greater improvement than the $svdCUIP$ based personalized search. In this experiment, the most improvement was obtained when the value of k for the $svdCUIP$ and $modSvdCUIP$ was 90 and 100, respectively. Note that in a reduced space, the

4.4 Experimental Evaluation

performance of the *modSvdCUIP* based personalized search degraded below 0; this means that it performed worse than the baseline search. However, when k was set to 50 and above, it showed improved performance.

These results show that both methods benefited from the dimensional reduction step. In the following experiments, the value of k for the *svdCUIP* and *modSvdCUIP* was set to 90 and 100, respectively.

4.4.2.6 Determining the value of distinctness parameter, d , for the AOL data set

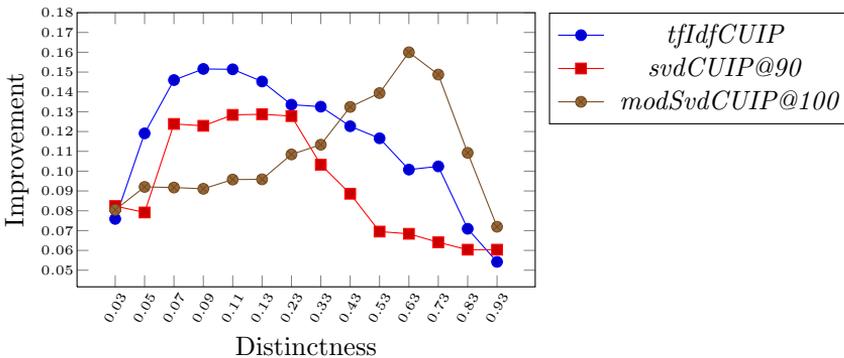


Figure 4.9: Estimating the values of distinctness parameter for *tfIdfCUIP*, *svdCUIP@90*, *modSvdCUIP@100* using Improvement as an evaluation metric.

The distinctness parameter d , controls how distinct or well separated the clusters are. As the value decreases, we get closer to a single cluster or a few large clusters; hence, grouping unrelated terms together or spanning multiple topic

4.4 Experimental Evaluation

areas. On the contrary, as the value increases, we end up with lots of clusters of a single term or lots of small-sized clusters, thus rendering the information in the clusters inadequate to represent topics. The parabolic graph in Figure 4.9 supports this idea. Note that there is no dimension reduction applied to the *tfIdfCUIP* method.

Figure 4.9 also shows that the *modSvdCUIP* based personalized search outperformed the *tfIdfCUIP* and *svdCUIP*. The maximum Improvement was obtained when d was set to 0.09, 0.13, and 0.63 for the *tfIdfCUIP*, *svdCUIP*, and *modSvdCUIP*, respectively. Performance of each CUIP is related to the number of clusters and the size of each cluster. The number of clusters for the *tfIdfCUIP* with $d=0.09$ is 54, 89 for the *svdCUIP@90* with $d=0.13$, and 76 for the *modSvdCUIP@100* with $d=0.63$. Also, the average number of tags in each cluster, average cluster size, for the *tfIdfCUIP* with $d=0.09$ is 6, 3 for the *svdCUIP@90* with $d=0.13$, and 4 for the *modSvdCUIP@100* with $d=0.63$. In short, having too many clusters, with only a few tags in each cluster, does not help disambiguate topics; this justifies why the *tfIdfCUIP* and the *modSvdCUIP* performed better than the *svdCUIP*.

In the following experiments that will execute on the evaluation data set, the value of d was set to 0.09 for the *tfIdfCUIP*, $k=90$ and $d=0.13$ for the *svdCUIP*, and $k=100$, $d=0.63$ for the *modSvdCUIP*.

4.4.2.7 Time to generate *svdCUIP* and *modSvdCUIP*

The aim of the experiment is to learn how much average time it takes to generate a *CUIP*. The results are plotted in Figure 4.10.

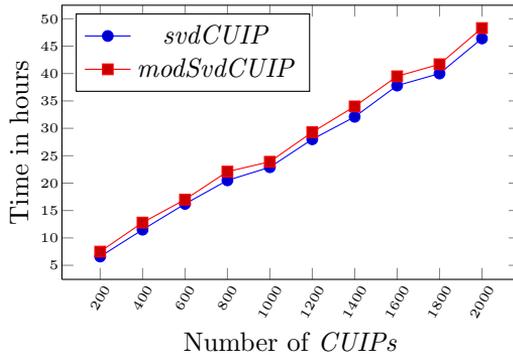


Figure 4.10: Average time to generate *svdCUIP* and *modSvdCUIP*

It shows that time to generate *CUIPs* is linear in nature. It took 46.4 and 48.3 hours to generate 2000 *svdCUIPs* and *modSvdCUIPs*, respectively, one for each individual user. In other words, a *svdCUIP* for a user can be generated in 83.52 sec, whereas a *modSvdCUIP* for a user can be generated in 86.94 sec. The difference is not huge. Note that, the generation of a *CUIP* is a background process so effectively it doesn't hurt the on-line execution time. Moreover, the time to generate a *CUIP* can be exponentially scaled down by using Mahout API that executes HAC on a hadoop cluster. We have already taken this viewpoint into consideration, therefore, since beginning all data at various stages is stored in csv file format.

4.4.2.8 Comparison of the *svdCUIP*, *modSvdCUIP*, and *tfIdfCUIP* for different classes of queries

The purpose of using the *modSvdCUIP* for personalized search is to identify the query context that we supposed the *tfIdfCUIP* would not be able to provide. However, the results presented in the previous sections indicate that the personalized search based on the *modSvdCUIP* and *tfIdfCUIP* delivered comparable effectiveness in improving the ranks of target Web documents. To further look into the effect that clusters have on personalized search, we analyzed the test collection, and found that self-evident queries didn't require disambiguation, and some vague queries received benefit when contextual tags were conflated with them. We identified 40 vague queries and 50 self-evident queries (refer to Appendix .1). Appendix .2 shows some examples of expanded queries and how query disambiguation is useful to personalized search.

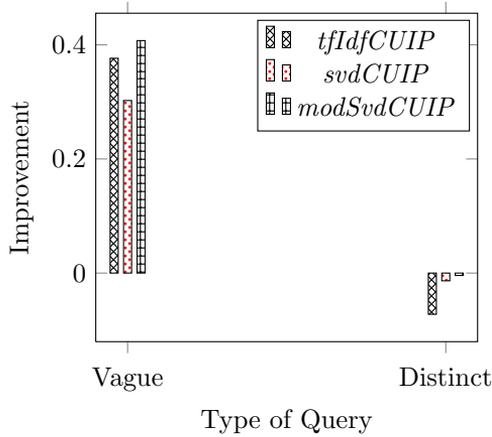


Figure 4.11: Comparing the Percentage Increase of the *tfIdfCUIP*, *svdCUIP*, *modSvdCUIP* for two classes of queries: vague and self-evident.

Figure 4.11 shows that the *modSvdCUIP* performed significantly better than both methods for the vague queries. And any modification of the self-evident queries by query expansion degraded the performance of the *CUIP* based personalized search methods. The *tfIdfCUIP* had the worst negative effect when used for disambiguating self-evident queries because the average cluster size is larger compared to other methods, thus degrading the ranks of the target Web documents.

4.4.2.9 Comparing all five methods - Improvement

This experiment aims to compare our proposed two methods with the others:
 1) tf based personalized search, *tfUIP*; 2) tfIdf based personalized search, *tfId-*

fUIP; and 3) *tfIdfCUIP* based personalized search.

As shown in Figure 4.12, the worst performer is the *tfIdfUIP*, similar to as reported by (David, Iván, and Joemon, 2010); results of both this study and (David, Iván, and Joemon, 2010) contradict those of (Xu, Bao, Fei, Su, and Yu, 2008) that the *tfIdfUIP* performed better than the *tfUIP*. A possible reason for the contradiction between ours and (Xu, Bao, Fei, Su, and Yu, 2008) approach is the total size of the result set; (Xu, Bao, Fei, Su, and Yu, 2008) re-ranked the top 100 Web documents, whereas our methods calculated the re-rank of the target URL in the top 600 documents. We suppose that the *tfUIP* showed better improvement than the *tfIdfUIP* because of the exclusion of two factors from the similarity score computation: document length and user profile length normalization factors. The user profile length normalization factor is dominant in the *tfIdfUIP*, and this penalizes the re-ranking score extensively.

The maximum improvement of the *modSvdCUIP* was 0.176766, whereas for the *svdCUIP* and the *tfIdfCUIP* was 0.132146 and 0.155571, respectively.

We performed significance test to determine if the difference between observed values from each approach are significant when compared with the baseline search. We used paired sample t-test and compared the average MRR values. Table 4.5 shows that the differences between the values from the *tfIdfUIP*, *tfUIP*, *tfIdfCUIP*, *svdCUIP*, and *modSvdCUIP* are significantly better than the baseline search. The MRR values were confirmed to be significantly different using the paired t-test with 95% confidence interval: *tfIdfUIP*(p-value=1.87E-09), *tfUIP* (p-value=1.67E-10), *tfIdfCUIP*(p-value=4.1E-

4.4 Experimental Evaluation

11), *svdCUIP* (p-value=4.2E-10), *modSvdCUIP* (p-value=2.31E-12). Thus, we can confidently conclude that the improvement of our proposed approaches is better than the baseline search.

	<i>tfIdfUIP</i>	<i>tfUIP</i>	<i>tfIdfCUIP</i>	<i>svdCUIP</i>	<i>modSvdCUIP</i>
MRR	0.3434	0.3625	0.4118	0.3946	0.4243

Table 4.5: Comparing the MRRs of *tfIdfUIP*, *tfUIP*, *tfIdfCUIP*, *svdCUIP*, and *modSvdCUIP*

4.4.3 Discussion

The strength of personalized search based on a *modSvdCUIP* lies in the discovery of second order similarity between tags, which is credited to the *modSim* tag-tag similarity matrix. The modSvd method generates a *modSvdCUIP* by applying HAC algorithm on the *modSim* matrix, which aids in discriminating tag sense by clustering semantically related tags together regardless whether they were originally collocated or not. Each cluster is assumed to correspond to a topic or to a sense of ambiguous tags. The poor result of personalized search based on *svdCUIP* is because it generated many small-size clusters resulting in inadequate disambiguation of user queries.

The best performance of *modSvdUIP* for the custom data set was observed when the dimension parameter k was set to 30. The average document space of the custom data set is 300, which is the average number of Web documents clicked by the users, and a reduced dimension space of 30 results in better

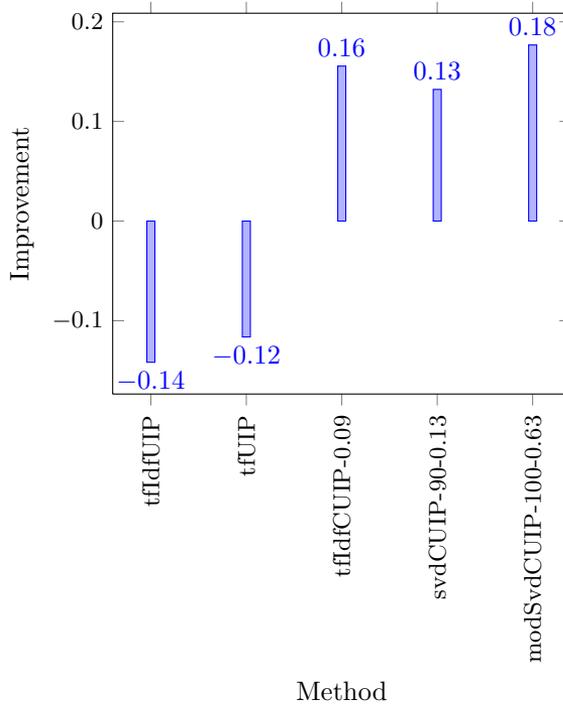


Figure 4.12: Comparing the Improvement of $tfIdfUIP$, $tfUIP$, $tfIdfCUIP-0.09$, $svdCUIP-90-0.13$, $modSvdCUIP-100-0.63$

performance. The best performance of $modSvdCUIP$ for the AOL query data set was observed when the dimension parameter k was set to 100. The average document space for the AOL data set is 500, significantly more than that of the custom data set. These results shows that the $modSvdCUIP$ was benefited from the dimensional reduction step. The $svdCUIP$ based personalized search also benefits from the dimension reduction step. For both data sets, the best performance was achieved when k was set to 90. We can draw the conclusion

4.4 Experimental Evaluation

that both approaches profited from the dimension reduction step. However, due to some small values in the similarity matrix *Sim*, the HAC algorithm couldn't clearly distinguish clusters that resulted in many small-size clusters, i.e., a topic is divided among several clusters. This resulted in poor performance of *svdCUIP* based personalized search compared to *modSvdCUIP* based personalized search in which the *modSim* matrix has comparatively higher similarity values, enabling HAC to clearly distinguish the clusters.

What distinguishes *CUIP* based personalized search approaches with other works that use social bookmarking services for personalized search is that tags in a user's *UIP* are dealt locally, and tags that constitute a *CUIP* are part of the vocabulary of a community of users who have annotated the documents clicked by the user. Tags in a user's *UIP* constructed based on (Noll and Meinel, 2007), (Xu, Bao, Fei, Su, and Yu, 2008), and (David, Iván, and Joemon, 2010) approaches are those used by the user to annotate documents of interest. As mentioned in the related work, there is a discrepancy between the vocabulary a user sees to formulate a search query and the vocabulary used in Web documents. Using only the user vocabulary to construct a *UIP* suffers from incomplete, insufficient tags. Building a user's *UIP* with tags that encompasses the world view can surpass this limitation to a certain extent.

(Noll and Meinel, 2007) doesn't include user and resource length normalization factor in the computation of cosine similarity score formulae. They neither normalize user profile tag frequencies nor resource profile tag frequencies; the tag weight of tags in the *UIP* is calculated by accumulating the count of tags,

4.4 Experimental Evaluation

and the term weight of terms in the resource profile is set to 1, if the term is used for annotating a document, else 0. This would allow equal importance to all documents and to all users. It makes sense not to normalize the tag weight of tags in user profile, because the terms were those that the user scribbled him (her)-self to annotate the documents. Xu's et al, on the other hand, use user and document length normalization factors resulting in the degradation of personalized search performance. Vallet et al. follows the same philosophy of Noll et al, and they adapt the Xu's approach by eliminating the user and document length normalization factor. Their justification for exclusion of normalization factor is similar to Noll's work that using the document length normalization factor would penalize the score of popular documents. Note that, similar to Noll's work, their approach also use all the tags in the *UIP* to compute the similarity score for re-ranking documents. Also, the similarity function computes the vector product of $t_{fu} * i_{uf}$ and $t_{fd} * i_{df}$ to calculate the similarity between *UIP* and document, where t_{fu} , t_{fd} , i_{uf} , and i_{df} is term frequency of a term in user profile, term frequency of a term in document profile, inverse user frequency, and inverse document frequency, respectively. Again, this kind of computation is only possible if we assume that every user who is searching the Web, (s)he is also actively tagging documents, otherwise how would one calculate i_{uf} . We present a more realistic approach, achieving a little better performance than (David et al., 2010), and making no assumptions about user's tagging activity. (Andriy, Jonathan, Bamshad, and Robin, 2008) presented a personalization algorithm for recommendation in folksonomies which relies on hierarchical tag

4.4 Experimental Evaluation

clusters. Note that the work is not about personalized search, but an adaptation of personalized search for recommendation of resources to the users of the folksonomy system based on their previous annotation of resources. Their approach clusters the entire tag space of the folksonomy system to obtain a common cluster structure to be used by all users of the folksonomy system. This approach is only applicable in a folksonomy system. Given a common cluster structure, tags in a user's *UIP* are mapped to appropriate clusters. It is like mapping a list of tags that have local scope to tag clusters that have global scope. This will augment the tags in the user *UIP*, thus encompassing the user's own vocabulary and of the community. A cluster structure will have all the possible semantic terms related to a topic. For example: consider a user's *UIP* has tags related to religion such as jewish, Israel, religion, etc (local scope). These tags will be mapped to a cluster that has the topic 'religion' in the common cluster structure (global scope). The mapped cluster may also have other tags related to religion such as Hinduism, Christianity, Buddhism, etc. Such kind of CUIP has properly identified the user general interests, for example, religion in this scenario, but it fails to identify the user's specific interests, which was originally jewish, but now after the CUIP is augmented, it also contains additional terms such as hinduism, buddhism, etc. To circumvent this limitation, (Andriy, Jonathan, Bamshad, and Robin, 2008) proposed to use three tuning parameters, step, generalization level, and division level, to limit the breadth of the mapped cluster. Our approaches also try to achieve the same objective, which is user oriented and bounded by the tags in the user's

4.4 Experimental Evaluation

UIP to generate a *CUIP*. However, we don't need any special parameters to limit the breadth of the cluster structure. This reduces the complexity and maximizes the accuracy of computing the cluster structure, also also increases the search quality. We also observed that not all queries benefit from the personalized search; the self-evident queries, also referred as navigational query (Broder, 2002), need not always be disambiguated, because the target Web documents for these queries are the same regardless of user interest. We found that applying personalized search to navigational queries reduces performance. The vague queries, which need to be disambiguated or could have different answers depending on user interests, benefit from the application of *CUIP* based personalized search.

One limitation of our proposed methods is that both the *UIP* and *CUIP* depend on the resource profile of resources. Tags in a *UIP* are aggregated from the resource profiles of Web documents. A resource profile for a Web document is only available if its annotations are available on a public social bookmarking service. We found for the AOL data set that approximately 34% of all the Web documents were not annotated on Delicious servers. Whereas, for the custom data set, 45% were not annotated on the same servers. One reason for this difference lies in the age of data sets: the AOL data set is older, hence there is a higher probability of the data being annotated. In our future work, we would like to experiment with OpenCalais¹ service for Web documents whose resource profiles cannot be constructed from social bookmarking services. OpenCalais

¹www.opencalais.com

4.4 Experimental Evaluation

returns topics, place names, people names, and URLs present in a document. This will also help us to develop a much better *UIP* and to improve the quality of personalized search.

Finally, the proposed methods can be used for personalizing search results generated from any search engines, and are very compatible for building a *UIP* or *CUIP* from any social bookmarking services. Our key contribution rests in developing a *CUIP*, and showing its usage for personalized search, one of many areas our methods can be applied for.

To conclude, the cluster structure emerging from a *modSvdCUIP* is able to identify user interests, group semantically related tags into clusters, identify second-order co-occurrence similarity between terms, and improve the search result quality. Personalized search based on *modSvdCUIP* performs better than approaches using the *tfUIP*, *tfIdfUIP*, and is comparable to the approaches *tfIdfCUIP* and *svdCUIP*. The improvement is due to the fact that the similarity matrix *modSim* is able to discover the sense of a topic by computing the first-order co-occurrence and second-order co-occurrence similarity between tags.

5

User Profiling for Partnership Match

In order to maximize the advantages and minimize the negative effects of globalization and growing interdependence, it is imperative for SMEs (Small and Medium Enterprises) in developing countries to forge partnerships with big enterprises in developed regions. However, the partnership establishment process is a rough ride; it comes with its own set of hurdles. A survey by PricewaterhouseCoopers (PwC) reveals that 44% of the partnerships were unsuccessful. We refer to research literature to find out various features that are involved during partnership establishment process. Based upon a review, we select features that form core concepts in a partnership establishment process. These concepts along with their related properties are modeled as an ontology, termed as Partnership Ontology. Big enterprises and SME (Small and Medium Enterprises) can use the partnership ontology to lay down their requirements as

a buyer profiles and seller profiles respectively. A semantic similarity measure is defined to compute a ranked list of matching seller profiles given a buyer profile. We illustrate the devised methodology of partnership establishment process by an example using a case study.

Globalization has ushered new gateways for SMEs in developing countries through greater integration into the world economy. The possibility to import new ideas, modern technology, and business investment opportunities from advanced countries can boost economic growth. Significant transfer of technology and modernization of the economies has occurred particularly in manufactured goods, through joint ventures, licensing agreements and other enterprise partnerships. Partnership is a voluntary collaborative agreement between two or more parties in which all participants agree to work together to achieve a common purpose or undertake a specific task which is a win-win situation for both. PwC interviewed CEOs of 239 Fortune 500 companies - results show that 56% of the companies in US have partnered over the past 5 years. These companies have partnered with large companies (41%), large MNCs (28%), large domestic companies (22%), small companies (29%), university (7%), and federal lab (3%).

A common theme among purchase managers from both failed and successful strategic alliances is the importance of building mutual trust and commitment among partners. No matter how mutually beneficial and logical the venture may seem without trust and commitment the alliance will fail entirely, or it will fail to reach its strategic potential. There are a variety of ways that a

company can attain and sustain commitment and trust in cooperative ventures. Goal and intent revelation is a crucial step toward building trust. The most common causes of failure ¹ cited by CEOs are: cultural differences (49%), poor or unclear leadership (49%) and poor integration process (46%). Though most enterprises understand and are aware of the reasons of the failure, they somehow fail to establish an amenable partnership. This is because they fail to spend enough resources understanding their individual needs and defining their requirements. As a result, there is a greater risk of an incorrect decision that ultimately leads to failed relationships

The projects that operate within inter-enterprise environments additionally face the problem that different information models are likely to be used by different partners. Engineers working within a particular organization will inevitably develop their own vocabulary for particular activities and these will need to be adjusted to be more practical and to meet the requirements of different collaborating partners. Hence, when two different partners are brought together, two common types of problem can occur in communications that share and exchange information, firstly, the same term is being applied to different concepts (semantic problem), secondly, different terms may be used to denote the same entity (syntax problem). This problem is popularly known as integration problem (Giachetti, 2004) in literature. Employment of ontology in this work resolves the integration problem. Thus a critical question is, how geographically separated organizations can be supported to establish a part-

¹http://www.1000ventures.com/business_guide/partnerships_main.html

nership that increases the probability of success?

In the previous two chapters, chapter 3 and chapter 4, I have presented how feature based user profiling can be used for building *UIPs* and *CUIPs*. In chapter 3, the feature anchor text of clicked Web pages by the user was used for building *UIPs*. In chapter 4, the feature tag annotations by a community of users to the clicked Web pages by the user was used for building *UIPs* and *CUIPs*. In this chapter, the features that are targeted are user preferences and context of work, refer Figure 1.1. A user explicitly input his preferences (attribute values) about the attributes of interest. Attributes are predefined and modelled as concepts in an ontology representing the context of work. This chapter also demonstrates how a buyer profile or seller profile is constructed by explicitly requesting a user to input his preferences about the concepts defined in the ontology, and how similarity is computed between different types of profiles. This chapter makes the following contributions:

1. I survey the research literature to identify the key concepts that are negotiated during a partnership establishment process.
2. Based upon the concepts identified in the previous step, an ontology is proposed, termed as Partnership Ontology.
3. Using Partnership Ontology, a manifestation of user profiles is illustrated as buyer profiles or seller profiles.
4. A semantic similarity match is proposed that recommends matching seller profiles for a given buyer profile.

5.1 Supplier Selection

In the traditional Supplier Selection process, an enterprise scrutinize potential suppliers from a given list of suppliers. An enterprise select potential suppliers from its previous dealings. A RFQ (Request For Quotation) is sent to all the potential suppliers. After receiving quotes from suppliers and based on the various other information listed in Table 5.1, an optimal supplier is selected. The whole process of supplier selection can be summarized into 6 steps:

1. Select Candidate Suppliers
2. Send RFQ (Request for Quotation)
3. Receive Quotations
4. Select Supplier
5. Negotiation
6. Signing the Contract

Though the above 6 step process for Supplier Selection looks trivial, it is a very time consuming and complex process. We list the various complexities that one encounters and side by side explain how our system deal with them.

1. To select potential suppliers, a buyer use the previous history or its dealings with the suppliers. This limits the number of supplier and hence

5.1 Supplier Selection

lower the competitiveness of the supplier selection process. New suppliers, who have had no interaction with the current buyer but have successful partnerships with other buyers, are not given due consideration. In-order to remove any biases, our system allows all suppliers to model their facilities or services as a seller profile.

2. Sending RFQ and receiving quotations is a time consuming process. Moreover RFQs are best suited to standardized products or services so that various supplier quotes can be easily comparable. This is a serious limitation which limits a system applicable to only a particular domain. The proposed system uses UNSPSC ontology ¹ for disambiguation of any product or services. The UNSPSC provides an open, global multi-sector standard for efficient and accurate classification of products and services. Using UNSPSC codes throughout an extended supply chain - seller, buyer, and distributor can process transaction data automatically and can perform management, analysis and decision function in time-critical ways that would not be possible without the codes. Classifying products and services with a common coding scheme facilitates commerce between buyers and sellers and is becoming mandatory in the new era of electronic commerce. Large companies are beginning to code purchases in order to analyze their spending. Classifying products and services with a common coding scheme facilitates commerce between buyers and sellers and is becoming mandatory in the new era of electronic

¹<http://www.ksl.stanford.edu/projects/DAML/UNSPSC.daml>

5.1 Supplier Selection

commerce. Large companies are beginning to code purchases in order to analyze their spending. The UNSPSC is designed to serve three primary functions: **Resource Discovery**, **Expenditure Analysis**, and **Product Awareness**. UNSPSC is a hierarchical classification having 5 levels, altogether it is a eight or ten digit numerical code. The codes are hierarchical, similar to an outline. As you get deeper in the outline, there is more detail. Each level contains a two character numerical value and a textual description. Based on this hierarchical structure, each UNSPSC code can be broken down as follows: the first 2 digits (from left) represent *segment*, next 2 digits represent *family*, next 2 digits represent *class*, second last 2 digits represent *commodity* and finally the last 2 digits are optional that represent *business function*. For ex., the UNSPSC code for Cooling or refrigeration services is 70142011 which is comprised of following categories. The *segment* code 70 for “Farming and Fishing and Forestry and Wildlife Contracting Services”, *family* code 14 for “Crop production and management and protection ”, *class* code 20 for “Post harvesting crop processing”, and finally the *commodity* code 11 for “Cooling or refrigeration services”.

3. An RFQ typical involves listing detailed specification of products or services. The more detailed the specifications, the more accurate the quote will be and comparable to the other suppliers. There is no standard for unit of measure and no distinct identifier for different product packaging

5.2 Criteria for Partnership Establishment

levels. For instance , one may order 20 and receive 200 because they are sold in units of 10. This results in inventories of wrong products and increased returns processing, driving up costs and creating cash flow issues. This work proposes a partnership ontology, that models the specifications as features and properties, also models unit of measurements similar to GoodRelations Ontology, refer (Hepp, 2008). Table 5.1 provides a snapshot of some of the important features that plays a key role for buyer - Supplier decisions are typically made following a comparison and analysis of the features.

5.2 Criteria for Partnership Establishment

The focus of work in this chapter provides a framework for establishment of buyer-seller partnership, where buyer are big enterprises and suppliers are SME (Small and Medium Enterprises). This section, in particular, investigates the core features or concepts required for building a profile i.e. the final goal results in a set of concepts and related properties that form an ontology for partnership establishment. The success of an establishment process is greatly reduced with the requirements criteria and their associated attributes being clearly known before the evaluation approach is implemented. In software engineering, requirement analysis encompasses those tasks that go into determining the needs of a customer. Requirement analysis determines the set of criteria to identify business needs i.e. what one party hopes to attain from another. The

5.2 Criteria for Partnership Establishment

complex process of partnership establishment generally involves assessing multiple criteria of varying importance, which may be quantitative or qualitative, tangible or intangible and which may involve trade-offs. (Dickson, 1966) and (Weber, Current, and Benton, 1991) provides a list of criteria that SMEs or enterprise negotiate over. Some of these criteria have gone obsolete over time due to changing business needs; therefore, we augment this list according to current requirements of partnership establishment process, refer Table 5.1. For example consider a scenario where a partnership under consideration between two geographically separated organizations, say one in USA and other one in Vietnam. Both partners have a different motivation for forging a partnership; an SME in Vietnam may be interested in a partnership so that they could learn advance technology whereas an organization in USA may be interested because of cheap labor costs. Since their motivations are different their requirements must also be different. Some of the other important criteria are discussed below. Financial Stability is one of the core requirements of a buyer; a SME with lot of debts can run the project into trouble. A match much be drawn between buyer requirements and seller manufacturing skills. Research and Development R&D includes assessing a potential partners level of R&D investment, the number of personnel involved in R&D, the communication network in place, the skill level of R&D personnel, and whether or not the organization engages in developing new products, and product and process improvement. A strong R&D presence in a potential partner organization is a positive sign for partnership. The next criterion is market knowledge and marketing skills, which involves

5.2 Criteria for Partnership Establishment

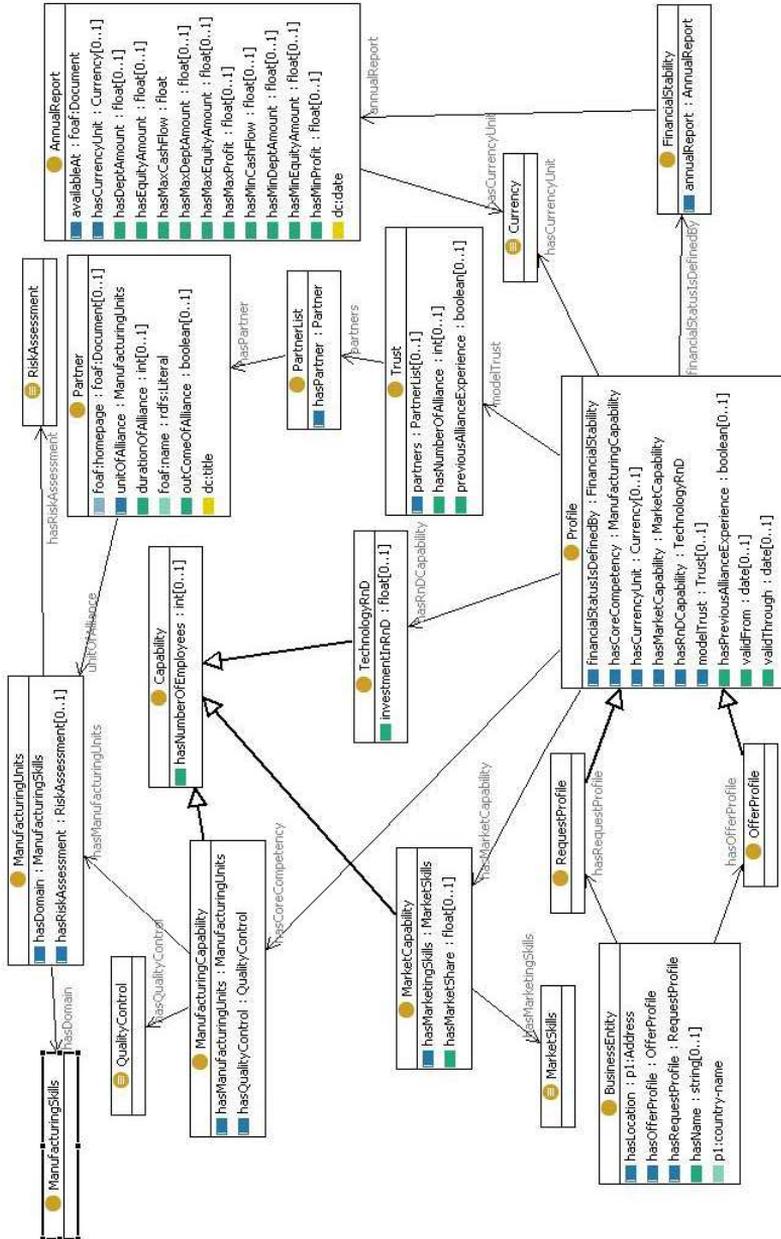


Figure 5.1: Partnership Ontology: concepts and properties that define relationship between them. Various other standard ontologies like Dublin Core, FOAF, Geo, VCard etc are also imported.

assessing the potential partners' market presence and understanding of both their competitors and customers. Alignment between the cultures of the SMEs and potential partner includes examining the cultural understanding between both organizations and their individual practices and behavior. A partnership often involves give and take or learning from each other, the willingness to share expertise criteria captures the notion of compatibility. One of the major criteria for forging partnership is trust which can be modeled using previous alliance experience. However, we strongly feel that trust should have more concrete concepts, therefore we have added more concepts under trust to model it comprehensively.

5.3 Partnership Ontology

In the following, we give an overview of the relevant conceptual entities and types of relationships. A definition of ontology by (Fensel) describes it as “specifically machine-readable information whose meaning is well defined by standards, which absolutely needs the inter-operable infrastructure that only global standard protocols can provide”. The concept involves categorizing structured and semi-structured information in a standard manner in order to give it meaning so that machines can understand it, process it and hence derive additional information, if any. Partnership ontology in Figure 5.1 is formulated from the concepts in Table 5.1; explained below are some additional concepts and properties that explain the relationship between them.

5.3 Partnership Ontology

In-order to build a common terminology for both enterprises and SMEs most of the concepts are modeled as enumeration. For ex: the concept **currency** is modeled as enumeration with two values USD and EURO; thus any concept that link to currency can only use USD and EURO as values. The partnership ontology is centered around concept **Profile**. Every **Business Entity** that wish to use this ontology must define a **Profile**. A **Profile** can be either **Buyer Profile** or **Seller Profile**. A concept **Profile** is modeled as a super concept of concept **Buyer Profile** and **Seller Profile** and all the properties are defined on concept **Profile**. Because of entailment rules, all the properties defined on concept **Profile** are inherited by both sup concepts **Buyer Profile** and **Seller Profile**. The concept **Profile** has properties that are instrumental in defining profiles; for ex: properties *financialStatusisDefinedBy*, *hasCapability*, *hasCoreCompetency* can define a user's profile financial status, manufacturing skills, manufacturing units and core competency respectively. Every profile has a validity duration which is modeled using two data type properties *validFrom* and *validThrough*.

The concept **FinancialStability** uses the concept **AnnualReport** to define an enterprise financial conditions and both concepts are related together using the property *annualReport*. The concept **AnnualReport** define various properties that can help where the annual report document can be located (*avaiableAt foaf:Document*), how much is the debt amount(*hasDeptAmount*), how much is the liquidity(*hasEquityAmount*), how much is the cash flow (*hasCashFlow*).

5.3 Partnership Ontology

The concept **Capability**, defines the core strength of an organization, is a super concept of three concepts **ManufacturingCapability**, **MarketCapability**, and **TechnologyRnD**. Note that, concepts **Manufacturing Skills** and **Manufacturing Facility** are enumerations. To model trust, which is a very essential part in any partnership establishment, we use the past history of alliances. A SME is trustworthy if he/she has successfully executed projects in partnership with other enterprises. Therefore, the concept **Trust** has a property *partners* which connect to concept **PartnerList**. Using the concept **PartnerList**, a number of partners can be defined, and each partner is modeled using the concept **Partner**. A partner is identified using the properties *foaf:homepage* and *foaf:name* to name a few. A concept **Partner** also contains information about domain of alliance modeled using property *unitOfAlliance* connected to concept **ManufacturingUnit** which can be further narrowed down to a particular manufacturing skills using the property *hasDomain*. The range of property *hasDomain* is **ManufacturingSkills** which represent the core service area. There can be various approaches to modeling **Manufacturing Skills**. The simplest approach could be instances of concept **Manufacturing Skills** be string literal which can create disambiguation, for ex: if a user uses a string value "Refrigeration", this has several further variations like "Industrial Refrigeration", "Cooling and Refrigeration Services", etc. It may be possible that engineers working at different organizations have different vocabulary - this would seriously effect the similarity match results of profiles. We propose to use UNSPSC web service, as described in section 3,

for disambiguation of **ManufacturingSkills**. Given a string literal, our system search its matching standard terms in the UNSPSC and return them in order of relevance. For ex: for string literal refrigeration, four matching terms are returned “Industrial refrigeration ”, “Cooling or refrigeration services ”, “HVAC refrigeration construction service ”, and “Air conditioning or ventilating or refrigeration equipment manufacture services”. Note that UNSPSC also returns the unique UNSPSC codes for each of the term. These standard codes are stored as an instance of **ManufacturingSkills**. Each manufacturing unit also contains information about risk assessment i.e. if an enterprise has implementation of risk assessment guidelines in their factory or workplace.

Another important concept for forging partnerships is partner marketing skills. This is modeled using the concept **MarketCapability** which is related to concept **Profile** using the property *hasMarketCapability*. The concept **MarketCapability** models the market skills and market knowledge of an SME using the properties *hasMarketSkills* and *marketKnowledge* respectively which are further related to enumerated concepts **Market Share** and **MarketSkills**. Concepts and Sub-concepts henceforth will be referred to as attributes and concept instances will be referred as attribute values.

5.4 Case Study

Most of the research work in the domain of Partnership Establishment takes a manual approach; asking purchase managers who participate in the study to

evaluate suppliers on a set of features and some sort of scale. It is important to note that, such a study only provides a subjective view of a set of managers and it would be inappropriate if their evaluation be generalized for the whole population. Therefore, the work in this chapter takes a personalized view - we ask the suppliers or sellers and buyers to provide their information and services respectively as a profile. We evaluate five candidate suppliers and one buyer using partnership ontology and semantic similarity measure. One Buyer profile and five supplier profiles are shown Figure 5.2, 5.3, 5.4, and 5.4. The information about suppliers and buyers were provided by the **Trade Investment Agency** (name withheld due to privacy issues). The provided information was then represented using partnership ontology.

5.4 Case Study

Name: seller1		Name: seller2	
Current Profits	100K-999K	Current Profits	1-99K
Manufacturing Facilities	ISO 6 class clean room	Manufacturing Facilities	ISO 6 class clean room
Growth Potential	> 1 Million	Growth Potential	
Cash Flow	100K-999K	Cash Flow	100K-999K
Human Resources	> 500 employees	Human Resources	100-499 employees
Currency	Euro	Currency	USD
Equity	100K-999K	Equity	
Technology R&D	Information management	Technology R&D	Information management
Debt Amount	1-99K	Debt Amount	
Unique Competency	Automotive manufacturing	Unique Competency	Automotive manufacturing
Management	greater 10 years management experience	Management	
Manufacturing Skills	Electronics	Manufacturing Skills	
Quality Control	ISO 14001:2004 certification	Quality Control	ISO 14001:2004 certification
Collaboration		Collaboration	
Future Capabilities		Future Capabilities	
Market Knowledge	Good market knowledge	Market Knowledge	
Marketing Skills	Extensive marketing skills	Marketing Skills	Good marketing skills
Market Share	20 - 49% market share	Market Share	
Marketing Objectives		Marketing Objectives	
Market Gaps	luxury cars	Market Gaps	
Partnership Potential		Partnership Potential	
Trust		Trust	
Personal Rapport		Personal Rapport	
Commitment		Commitment	
Reputation		Reputation	
Dependancy		Dependancy	
Flexibility		Flexibility	
Cultural Alignment		Cultural Alignment	High level cultural alignment
Willingness to Share Expertise	High level willingness	Willingness to Share Expertise	
Previous Alliance Expertise	Yes	Previous Alliance Expertise	
Partnership Strategy		Partnership Strategy	
Shared Goals		Shared Goals	
Location		Location	
Political Links		Political Links	
Risk Assessment		Risk Assessment	
Relationship Maintenance		Relationship Maintenance	

Figure 5.2: Seller Profiles for this study: Seller1 and Seller2

5.4 Case Study

Name: seller3		Name: seller4	
Current Profits	1-99K	Current Profits	1-99K
Manufacturing Facilities	ISO 6 class clean room	Manufacturing Facilities	ISO 6 class clean room
Growth Potential	100K-999K	Growth Potential	100K-999K
Cash Flow	100K-999K	Cash Flow	100K-999K
Human Resources	50-99 employees	Human Resources	50-99 employees
Currency		Currency	USD
Equity	100K-999K	Equity	100K-999K
Technology R&D	CAD	Technology R&D	CAD
Debt Amount	100K-999K	Debt Amount	100K-999K
Unique Competency	Automotive manufacturing	Unique Competency	Automotive manufacturing
Management		Management	1 - 4 years management experience
Manufacturing Skills	Electronics	Manufacturing Skills	
Quality Control	ISO 9001:2000 certification	Quality Control	ISO 9001:2000 certification
Collaboration		Collaboration	
Future Capabilities		Future Capabilities	
Market Knowledge		Market Knowledge	
Marketing Skills	Limited market skills	Marketing Skills	
Market Share		Market Share	
Marketing Objectives		Marketing Objectives	
Market Gaps		Market Gaps	
Partnership Potential		Partnership Potential	
Trust		Trust	
Personal Rapport		Personal Rapport	
Commitment		Commitment	
Reputation		Reputation	
Dependency		Dependency	
Flexibility		Flexibility	
Cultural Alignment		Cultural Alignment	
Willingness to Share Expertise	High level willingness	Willingness to Share Expertise	High level willingness
Previous Alliance Expertise		Previous Alliance Expertise	Yes
Partnership Strategy		Partnership Strategy	
Shared Goals		Shared Goals	
Location		Location	Korea
Political Links		Political Links	
Risk Assessment		Risk Assessment	
Relationship Maintenance		Relationship Maintenance	

Figure 5.3: Seller Profiles for this study: Seller3 and Seller4

5.4 Case Study

Name: seller5	
Current Profits	1-99K
Manufacturing Facilities	10-499 sqm
Growth Potential	
Cash Flow	1-99K
Human Resources	
Currency	Euro
Equity	1-99K
Technology R&D	Robotics
Debt Amount	
Unique Competency	Automotive manufacturing
Management	
Manufacturing Skills	Welding
Quality Control	ISO 9001:2000 certification
Collaboration	
Future Capabilities	
Market Knowledge	
Marketing Skills	
Market Share	
Marketing Objectives	
Market Gaps	
Partnership Potential	
Trust	
Personal Rapport	
Commitment	
Reputation	
Dependency	
Flexibility	
Cultural Alignment	
Willingness to Share Expertise	
Previous Alliance Expertise	
Partnership Strategy	
Shared Goals	
Location	
Political Links	
Risk Assessment	

Figure 5.4: Seller Profiles for this study: Seller5

Attribute Manager

Request Name : AutomotiveRequest

Attribute Library	Attribute Values
<ul style="list-style-type: none"> Current Profits Manufacturing Facilities Growth Potential Cash Flow Human Resources Cash Flow Equity Technology R&D Debt Amount Unique Competency Management Manufacturing Skills Quality Control Collaboration Future Capabilities Market Knowledge Marketing Skills Market Share Marketing Objectives Market Gaps Partnership Potential Trust Personal Rapport Commitment Reputation Dependency Flexibility Cultural Alignment Willingness to Share Expertise Previous Alliance Expertise Partnership Strategy Shared Goals Location Political Links Risk Assessment Relationship Maintenance 	<div style="border: 1px solid blue; padding: 5px;"> <p>Marketing Skills Good marketing skills</p> <p>Risk Assessment Medium risk</p> <p>Partnership Potential Good partnership potential</p> <p>Technology R CAD</p> <p>Equity > 1 Million</p> <p>Cash Flow 100K-999K</p> <p>Human Resources > 500 employees</p> <p>Manufacturing Facilities ISO 6 class clean room</p> <p>Current Profits 100K-999K</p> <p>Willingness to Share Expertise High level willingness</p> <p>save</p> </div>

Figure 5.5: An example to demonstrate construction of user profile (Buyer Profile) - concepts shown here are derived from the Partnership Ontology

5.4.1 Buyer Profile and Seller Profile

The success of partnership establishment is significantly influenced by the manner in which profiles are constructed. A profile is simply a set of generic facts about a company, which may be used by other companies to determine their suitability as potential partners. A seller profile is a mechanism utilized to communicate what the potential partner can do to meet their needs. A seller profile records the capabilities and services that he has for offer. A buyer profile is a mechanism utilized to communicate the expectations that an enterprise has from a potential partner. Both the profiles are generated using the Partnership Ontology introduced in Section 5. An enterprise (henceforth called as buyer) looking for partners makes a buyer profile, whereas, SMEs make a seller profile. Note that both are oblivious of each other, i.e. they just make their profiles available to the system. Buyer, after providing his profile to the system, searches for the matching seller profiles, which the system returns after executing a semantic similarity match among various seller profiles available to the system. The result from searching is a set of possible partners that a buyer can consider to be his/her future partners. We developed a web service, that uses Partnership Ontology to construct seller profile and buyer profile using the Partnership Ontology, termed as e-Partner. This web service is developed using Java technologies, AJAX, Java Script and HTML. The web-service is available on-line and accessible through the following URL <http://tinyurl.com/yau5mfg>. Figure 5.6 and Figure 5.5 shows an exemplary use of web service to create a

Buyer Profile or Seller Profile.

After building a buyer profile, an enterprise can search for matching seller profiles by using the search functionality. But, before using the search option, a buyer can set the weights for the attributes which associates importance to the attributes, refer Figure 5.6. The weight assigned to attributes signifies the importance of the attribute and is used in the calculation of similarity distance i.e. if a particular attribute in a buyer profile has weight 0.5 and the same attribute is also present in a seller profile, its similarity score will be greater, however if it is absent in a seller profile then similarity score for that particular attribute will be 0. The knock-out property selected for a particular attribute in a buyer profile can be interpreted as follows; if a seller profile does not has that attribute in its profile, simply discard the profile. In other words, knock out property makes an attribute essential and puts a restriction that a prospective seller has to have that attribute in its profile. A sourcing property for a particular attribute if checked signifies that this particular attribute is insignificant. In other words, if an attribute, is checked for sourcing property in a buyer profile and, is missing from a seller profile, it will still be considered for calculating the overall similarity score. For instance, if a buyer profile has 3 attributes a_1 , a_2 , and a_3 , and a seller profile has 2 attributes a_1 and a_2 , this evaluates to 66.67% similarity, but, if a buyer profile has the sourcing property selected for a_3 , similarity score will now evaluate to 100%. Note that similarity score of any 2 attributes also depends on the depth of attribute values. The sourcing property is included for experimentation, so that a buyer can actually

evaluate how many sellers show up if they unselect a particular attribute. Also note that, weight, sourcing and knock-out properties are not available for a seller profile.

5.4.2 Semantic Similarity Measure

Given a collection of buyer profiles and seller profiles, the next step would be to find a ranked list of seller profiles for a given buyer profile. In order to compute a ranked list, we propose a semantic similarity measure which is motivated from (Salton, Wong, and Yang, 1975) work on Vector Space Model. First, we briefly explain what is vector space model and how it can be modelled to suit our needs. Following it, we postulate two definitions to lay the basis for mathematical formulate for computation of similarity measure of profiles.

5.4 Case Study

Name: AutomotiveRequest				
Competencies	Value	Weight	Sourcing	Knockout
<u>Current Profits</u>	100K-999K	0.5	<input type="checkbox"/>	<input type="checkbox"/>
<u>Manufacturing Facilities</u>	ISO 6 class clean room	0.2	<input type="checkbox"/>	<input type="checkbox"/>
<u>Growth Potential</u>		0.1	<input type="checkbox"/>	<input type="checkbox"/>
<u>Cash Flow</u>	100K-999K	1	<input type="checkbox"/>	<input type="checkbox"/>
<u>Human Resources</u>	> 500 employees	1	<input type="checkbox"/>	<input type="checkbox"/>
<u>Currency</u>	USD	0.1	<input type="checkbox"/>	<input type="checkbox"/>
<u>Equity</u>	> 1 Million	0.8	<input type="checkbox"/>	<input type="checkbox"/>
<u>Technology R&D</u>	Information management	0.1	<input type="checkbox"/>	<input type="checkbox"/>
<u>Debt Amount</u>	1-99K	0.8	<input type="checkbox"/>	<input type="checkbox"/>
<u>Unique Competency</u>	Automotive manufacturing	1	<input type="checkbox"/>	<input checked="" type="checkbox"/>
<u>Management</u>	High level management support	0.5	<input type="checkbox"/>	<input type="checkbox"/>
<u>Manufacturing Skills</u>	CNC machining	1	<input type="checkbox"/>	<input type="checkbox"/>
<u>Quality Control</u>	ISO 14001:2004 certification	0.8	<input type="checkbox"/>	<input type="checkbox"/>
<u>Collaboration</u>		0.1	<input type="checkbox"/>	<input type="checkbox"/>
<u>Future Capabilities</u>		0.1	<input type="checkbox"/>	<input type="checkbox"/>
<u>Marketing Skills</u>	Good marketing skills	0.6	<input type="checkbox"/>	<input type="checkbox"/>
<u>Market Share</u>	20 - 49% market share	0.6	<input type="checkbox"/>	<input type="checkbox"/>
<u>Cultural Alignment</u>	High level cultural alignment	0.1	<input type="checkbox"/>	<input type="checkbox"/>
<u>Willingness to Share Expertise</u>	Medium level willingness	0.1	<input type="checkbox"/>	<input type="checkbox"/>
Initiate Search		Show Results		
		Top 5		

Figure 5.6: A reduced version of buyer profile - truncated to fit in here. The features that buyer does not choose during profile construction are removed to save space.

VSM is a linear algebraic method most commonly used in Information Retrieval for representing text documents as vectors and aids in relevancy ranking of documents with respect to the inputted query. A document is represented as a vector in an m dimension subspace, where m constitutes the number of words in the dictionary. If a word or term occurs in the document, its value in the vector is 1 otherwise 0. Hence, such kind of vector tends to be sparse. Moreover, if we constitute a term-document matrix i.e. terms as rows and documents as columns, the matrix formed will be sparse matrix. Motivated by the terminology used in Vector Space Model, we would like to borrow it, improvise it and use it in the context of supplier match. Here, we define a *profile vector* and an *attribute-profile matrix* to suit Vector Space Model to our needs. The profile-attribute matrix will not be very high dimensional because in the current scenario attributes are finite as compared to terms in a dictionary which are infinite (or a very large number).

Definition 1: A *Profile Vector* $P^{(i)}$ is represented by a m -dimensional vector

$$P^{(i)} = \{att_1, att_2, \dots, att_m\} \quad (5.1)$$

where att_m , is a name of an attribute.

The actual *Profile Vector* P^i after substitution of values for attributes will be

$$P^{(i)} = \{av_{i1}, av_{i2}, \dots, av_{im}\} \quad (5.2)$$

where av_{im} is a value for att_m for profile $P^{(i)}$.

Definition 2: An *Attribute-Profile Matrix* is a mathematical matrix that describes the value of various attributes that occurs in a collection of profiles. Each column correspond to a profile in the collection, and each row corresponds to an attribute with its attribute-value.

$$A_{n,m} = \begin{pmatrix} av_{1,1} & av_{2,1} & \cdots & a_{n,1} \\ av_{1,2} & a_{2,2} & \cdots & a_{n,2} \\ \vdots & \vdots & \ddots & \vdots \\ a_{1,m} & a_{2,m} & \cdots & a_{n,m} \end{pmatrix} \quad (5.3)$$

Now, a column in the *Attribute-Profile Matrix* is a column vector corresponding to a profile, giving its relation to each attribute.

Given the profile vectors for two different profiles (of course, one is a buyer profile and other is a seller profile), it is possible to compute a similarity between them, $sim(P^i, P^j)$, which reflects the degree of similarity between two profiles. Such a similarity measure will be an inner product of the two vectors. When two vectors are identical, the cosine of angle between them will be 0, producing a maximum similarity.

Suppose, let us represent an exemplary profile vector according to definition 1 as $\{P^{(i)}; i=1, \dots, n\}$ of attributes of n different partners. A profile vector, $P^{(i)}$, will be represented in m -dimension subspace as a vector, where m -dimension subspace consists of m different attributes represented in space. Equation 4

shows a $1 \times m$ column matrix representation of profile vector (buyer or seller).

$$P^{(i)} = \begin{bmatrix} att_1 \\ att_2 \\ \vdots \\ att_m \end{bmatrix} \quad (5.4)$$

Or, a profile with attribute-values substituted for attributes will be

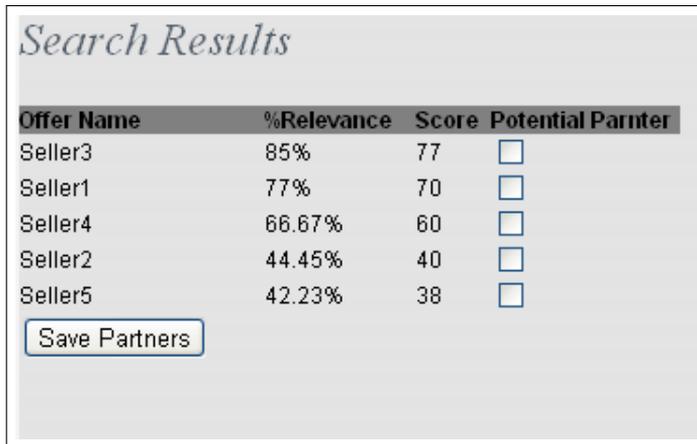
$$P^{(i)} = \begin{bmatrix} av_{i1} \\ av_{i2} \\ \vdots \\ av_{im} \end{bmatrix} \quad (5.5)$$

To compute the similarity of a buyer profile with seller profiles, we can take cross product of vector representation of buyer profile with various seller profiles using equation 6.

$$Sim(P^{(i)}, P^{(j)}) = \frac{\sum_{k=1}^m av_{ik} * av_{jk}}{\sqrt{\sum_{k=1}^m av_{ik}^2} * \sqrt{\sum_{k=1}^m av_{jk}^2}} \quad (5.6)$$

Equation 6 aids in generating a ranked list of seller profiles with respect to similarity of a buyer profile. Result of such a computation is a value between 0 and 1, where 1 signifies 100% match and 0 signifies no match, 0.5 signifies 50-50 match, and so on. A preview of search results is shown in Figure 5.7. Key

information provided in this view includes the seller name, percentage relevance of seller profile in relation to the buyer profile, check box for potential partner selection.



Offer Name	%Relevance	Score	Potential Partner
Seller3	85%	77	<input type="checkbox"/>
Seller1	77%	70	<input type="checkbox"/>
Seller4	66.67%	60	<input type="checkbox"/>
Seller2	44.45%	40	<input type="checkbox"/>
Seller5	42.23%	38	<input type="checkbox"/>

Save Partners

Figure 5.7: Search Results showing the ranked list of matching seller profiles to a given buyer profile.

5.5 Discussion

The process to establish a partnership is implemented and tested based on 1 buyer profile and 5 seller profiles. Buyer Profile in Figure 5.6, note that the feature **Unique Competency** has knockout attribute selected. This means, if any of the sellers do not have the feature **Unique Competency** in their seller profile or do not have the value “Automotive Manufacturing” for Unique Competency will be simply discarded. The sourcing attribute and knockout

attribute works exactly opposite of each other; one (knockout) is very strict whereas other (sourcing) is very lenient. Also, buyer1 has higher weight for following features **Cash Flow**, **Human Resources**, and **Manufacturing Skills** whereas the follower features has lower weight **Currency**, **Cultural Alignment**, and **Willingness to Share Expertise**. Higher weight for features suggests their importance and lower weight suggests that they are less important.

In this case study, all the seller profiles have the value “Automotive Manufacturing” for feature **Unique Competency** in their profile, so none of them is knocked-out. The seller with the highest score is regarded as the best performing seller and the rest can be ranked accordingly. The results, from case study, indicates that the top two sellers are seller3 and seller1 - their respective relevance percentages are 85% and 77%. We believe these sellers receive more business than any other seller, however, empirical studies have revealed that relevance score less than 50% reflects seller whose priorities do not align with buyer’s requirements. Semantic Similarity measure shows that Seller 4 is relatively better than Seller 2. For this work, we can regard 50% as cut off value. Note that, a buyer is choose to free the cut-off point, it can be a percentage relevance or top 5 or top10. He can then negotiate with the seller and further align their respective ambitions. The main advantages can be described as follows

1. The proposed methodology for partnership establishment allows selecting sellers in a global environment thus enables sellers to expand themselves

globally. The system provides an access point for buyers to source partners in globally disperse developed and developing countries. Therefore, it allows buyers to embark into emerging markets such as China, India and reduce their manufacturing costs, resources, and gain expertise.

2. Generating, storing, manipulating, and distributing information is central to a successful partner establishment process. The challenge of making relevant information available in distributed partnership establishment is addressed by Partnership Ontology. The problem of synonymy and polysemy is taken care of by the UNSPSC ontology. Ontology in this case allows machine readable representation of buyer profiles and seller profiles. Some of the other advantages that come with the use of ontologies is that they are easy to update, can easily borrow concepts and properties from other ontologies and expand themselves, can be merged together with other ontologies, etc.

5.6 Conclusions

Most of the research work in partnership establishment rank sellers, given buyer requirements. They use various mathematical models like AI, Neural Network, DES, Analytic Hierarchy Process (AHP), and Quality Function Deployment (QFD). To the best of our knowledge, no work exists that have addressed the integration problem in partnership establishment process. In this work, we capitalise on ontologies to provide a machine readable representation of buyer

and seller profiles, propose a semantic similarity measure to rank seller profiles for a given buyer profile. We also implemented a web service that automates the whole process from representation of profiles to final ranking of seller profiles. It is evident from the results, analysis and the discussion outlined in the previous sections that the methodology presented in this chapter is a feasible, useful and practical for ranking buyer-seller in a globalized situation. The proposed methodology is unique in the sense that ontologies are employed and vector space model is used so as to provide a solid systematic approach which is also mathematically proven. The major innovation of the proposed methodology is that the UNSPSC ontology provides a unique code for **manufacturing skills** that helps in disambiguation of any product or services. Classifying products and services with a common coding scheme facilitates commerce between buyers and sellers and is becoming mandatory in the new era of electronic commerce. There are some delicate issues like privacy, cultural, intellectual property rights, etc that needs to be addressed in this research. As a future work, this work can be extended for the ownership type partnerships or joint ventures etc. To extend this work, such that, multiple SMEs or partners be selected for a given job and how to distribute jobs among them is an interesting research problem

.

6

Conclusion

Now this is not the end. It is not even the beginning of the end. But it is, perhaps, the end of the beginning. - Winston Churchill

In this dissertation, I worked on different manifestations of user profile for different domains. In the domain of personalized search, a user profile is manifested as User Interest Profile (*UIP*) and Clustered User Interest Profile (*CUIP*). I proposed three novel methods that exploited user search history and social bookmarking services for building a User Interest Profile(*UIP*) and Clustered User Interest Profile (*CUIP*) that consists of term clusters of user interests. The first method for personalized search is termed as Exclusively Yours'. It builds a *UIP* from the anchor text of hub pages of the user clicked Web documents. We also proposed a method to calculate the term-weights that originates from multiple documents and are accumulated in the *UIP*. After the construction of a *UIP*, we propose a query expansion method that relies on information distance and discounts the terms that have not been updated for a time dura-

tion, thus, logically segregating a *UIP* into two parts. The proposed method is compared against non-folksonomy based personalized search methods and non-personalized search using the Precision, Discounted Cumulative Gain (DCG), and Average Rank (AR) evaluation metrics. It has demonstrated improved search quality against its comparators. The results were satisfactory but it has its own limitations. We found that a *UIP* constructed from anchor text also has some unintentional noise embedded into it.

The second method, to construct a *UIP* and *CUIP*, is based on the Singular Value Decomposition (SVD) to compute a tag-tag similarity matrix and use the Hierarchical Agglomerative Clustering (HAC) on the matrix to generate a cluster structure, *svdCUIP*. The third method is an extension of the first method, called modified Singular Value Decomposition (modSVD), that aims to group related tags based on their second-order co-occurrence similarity. This method is based on the assumption that related tags are often expressed together by similar sets of tags. These semantically related tags are bound to co-occur with similar neighbours. The objective of the modSVD is to discover and group these semantically related tags into clusters to generate a *modSvdCUIP*, each cluster of which identifies a unified topic. For these two methods, we proposed an automatic evaluation method that does not require user involvement to enumerate the relevancy of search results. We found out it to be an effective method to compare personalized search methods.

To evaluate the effectiveness of the proposed approaches, we compared them with the baseline search and the three other methods that use folksonomy for

constructing *UIP* and Resource Profile (*RP*): *tfUIP* (Noll and Meinel, 2007), *tfIdfUIP* (Xu, Bao, Fei, Su, and Yu, 2008), *tfIdfCUIP*. Our methods are more realistic as they make no assumption about the tagging activity of the user, and can be easily put to practice for any user who uses a search engine for his/her daily search needs. In our evaluations, we found that the improvement in the ranking scores of the target URLs for the *modSvdCUIP* based personalized search were better than all the other methods; the *modSvdCUIP* approach showed improvement of 71.6%, 27.8%, 12%, 6.6%, and 8.1% over the baseline (Lucene Search), *tfIdfUIP*, *tfUIP*, *tfIdfCUIP*, and *svdCUIP* approaches, respectively.

All three proposed methods are non-invasive. In other words, they make no attempt to collect user personal information. The only objective is to mine user interests and find relationship between them. Each cluster, in the cluster structure *CUIP*, identifies a distinct topic, and the application of *CUIP* aids in disambiguating the context of use query, which is particularly needed for vague queries. It is also very effective in disambiguating the synonymy and polysemy terms.

In the domain of Partnership Match, a user profile is manifested as a buyer profile or seller profile which is drawn from a controlled vocabulary. The controlled vocabulary in this case is an ontology. I also proposed an ontology, termed as partnership ontology, which contains the concepts and relationship between them. A semantic similarity measure based on Vector Space Model is proposed to score and rank seller profiles for a given buyer profile. To the

best of our knowledge, no work exists that have addressed the integration problem in partnership establishment process. The partnership ontology provides a machine readable representation of buyer and seller profiles. The proposed methodology is unique in the sense that ontologies are employed and vector space model is used so as to provide a solid systematic approach which is also mathematically proven.

6.1 Future Work

Last, but not least, several issues need to be targeted to improve the personalized search and partnership match. In the next two subsection, I talk about the future wor in the domain of personalized search and the last section is about partnership match.

6.1.1 Degree of Personalization

Experiment results in personalized search suggest that not all queries need personalization. One task that remains outstanding is how to determine which query needs personalization and which does not. This task can be, to some extent, tackled by classifying the nature of the queries(Broder, 2002): navigational, Informational queries, transactional queries. We also observed in our experiments that navigational queries do not need disambiguation. For instance, the topmost result for the query "jigsaw puzzle" is <http://www.zigzone.com>, which is the best possible match; the query "jigsaw puzzle" does not require

any disambiguation. However, information queries, for instance "puzzle game", that cover a broad range of topics can be benefited by personalization; part of the reason is user's inability to represent his information needs in 2 or 3 words (Amanda, Dietmar, Major, and Tefko, 2001), the average length of user's query. It is easy to determine the type of query by using statistical methods (Rose and Levinson, 2004) or using machine learning approaches (Beitzel et al., 2005). It is the need of the hour that a personalized search web service should automatically diagnose the nature of input query and decide if it needs to be disambiguated or not.

6.1.2 Filter Bubble

A contrarian view to personalized search is "**Filter Bubble**". According to Wikipedia¹, a filter bubble is a result state in which a search algorithm selectively guesses what information a user would like to be interested in based on interests of the user which are largely derived from the user past click behavior (search history), twitter posts, Web pages visited. Some of the examples are Google's Personalized Search, Facebook recommendations, twitter news recommendation, and so on. This term was coined by internet activist Eli Pariser in his book (Pariser, 2011) that states, "users get less exposure to conflicting viewpoints and are isolated intellectually in their own information bubble". In other words, the information bubble subdues serendipity which closes us off to new ideas, subjects, and important information. In my future work, I would like

¹http://en.wikipedia.org/wiki/Filter_bubble

to study the effect and magnitude of information bubble on personalization so that a quantifiable measure can be development to calculate the effect. This in turn might also provide directions in drawing a balance between personalization and information bubble.

I will also look into more advanced methods such as probabilistic LSI and Latent Dirichlet Allocation(LDA) for discovering and building a more efficient *CUIP*.

6.1.3 IPR issues in Partnership Match

One of the issues that needs to be addressed is intellectual property rights (IPR), it needs to be protected during the partnership establishment process. Several sophisticated methods for information exchange via the Internet are being developed, however, end users are reluctant to share their information on-line. For the future research, I would like to focus on how to embed trust in user profiles (buyer profile or seller profile) in the partnership match, and how to control access to information during partnership establishment.

Bibliography

- F. Abel, Q. Gao, G.-J. Houben, and K. Tao. Analyzing user modeling on twitter for personalized news recommendations. In *Proceedings of the 19th international conference on User modeling, adaption, and personalization, UMAP'11*, pages 1–12, Berlin, Heidelberg, 2011. Springer-Verlag. ISBN 978-3-642-22361-7. 5, 45
- D. Achlioptas and F. McSherry. Fast computation of low rank matrix approximations. In *Proceedings of the thirty-third annual ACM symposium on Theory of computing*, pages 611–618. ACM, 2001. 41
- E. Agichtein, E. Brill, and S. Dumais. Improving web search ranking by incorporating user behavior information. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval, SIGIR '06*, pages 19–26, New York, NY, USA, 2006. ACM. ISBN 1-59593-369-7. 43, 48, 87

BIBLIOGRAPHY

- F. Alan, K. Ravi, and V. Santosh. Fast monte-carlo algorithms for finding low-rank approximations. *Journal of the ACM*, 51(6):1025-1041, 2004. 41
- S. Amanda, W. Dietmar, B. J. J. Major, and S. Tefko. Searching the web: The public and their queries. *Journal of the American Society for Information Science and Technology*, 52:226-234, 2001. 168
- S. Andriy, G. Jonathan, M. Bamshad, and D. B. Robin. Personalized recommendation in social tagging systems using hierarchical clustering. In *RecSys*, pages 259-266, 2008. 45, 46, 47, 50, 86, 101, 129, 130
- C. Basnet and J. M. Leung. Inventory lot-sizing with supplier selection. *Computers & Operations Research*, 32(1):1-14, 2005. 52, 54
- O. Bayazit. Use of analytic network process in vendor selection decisions. *Benchmarking: An International Journal*, 13(5):566-579, 2006. 53, 144
- I. H. Beaumont. User modelling in the interactive anatomy tutoring system anatom-tutor. *User Modeling and User-Adapted Interaction*, 4(1):21-45, 1994. 4
- S. M. Beitzel, E. C. Jensen, O. Frieder, D. Grossman, D. D. Lewis, A. Chowdhury, and A. Kolcz. Automatic web query classification using labeled and unlabeled training data. In *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '05, pages 581-582, New York, NY, USA, 2005. ACM. ISBN 1-59593-034-5. 168

BIBLIOGRAPHY

- M. W. Berry, M. Browne, A. N. Langville, V. P. Pauca, and R. J. Plemmons. Algorithms and applications for approximate nonnegative matrix factorization. *Computational Statistics & Data Analysis*, 52(1):155–173, 2007. 42
- L. Bing. *Web Data Mining: Exploring Hyperlinks, Contents, and Usage Data (Data-Centric Systems and Applications)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006. ISBN 3540378812. 104, 106
- D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *the Journal of machine Learning research*, 3:993–1022, 2003. 42
- M. R. Bouadjenek, H. Hacid, and M. Bouzeghoub. Laicos: an open source platform for personalized social web search. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1446–1449. ACM, 2013a. 50
- M. R. Bouadjenek, H. Hacid, M. Bouzeghoub, and A. Vakali. Using social annotations to enhance document representation for personalized search. In *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval*, pages 1049–1052. ACM, 2013b. 50
- C. Boyle and A. O. Encarnacion. Metadoc: an adaptive hypertext reading system. In *Adaptive Hypertext and Hypermedia*, pages 71–89. Springer, 1998. 4

BIBLIOGRAPHY

- T. J. Brailsford, C. D. Stewart, M. R. Zakaria, and A. Moore. Autonavagation, links and narrative in an adaptive web-based integrated learning environment. 2002. 4
- G. Brajnik, G. Guida, and C. Tasso. User modeling in intelligent information retrieval. *Information Processing & Management*, 23(4):305–320, 1987. 6
- S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. *Comput. Netw. ISDN Syst.*, 30:107–117, April 1998. ISSN 0169-7552. 43, 57
- A. Broder. A taxonomy of web search. *SIGIR Forum*, 36:3–10, September 2002. ISSN 0163-5840. 131, 167
- P. Brusilovsky and D. W. Cooper. Domain, task, and user models for an adaptive hypermedia performance support system. In *Proceedings of the 7th international conference on Intelligent user interfaces*, pages 23–30. ACM, 2002. 6
- P. Brusilovsky, A. Kobsa, and W. Nejdl. *The adaptive web: methods and strategies of web personalization*, volume 4321. Springer, 2007. 1
- P. L. Brusilovsky. A framework for intelligent knowledge sequencing and task sequencing. In *Intelligent tutoring systems*, pages 499–506. Springer, 1992. 6
- W. Buntine. Variational extensions to em and multinomial pca. In *Machine Learning: ECML 2002*, pages 23–34. Springer, 2002. 42

BIBLIOGRAPHY

- W. Buntine and A. Jakulin. Discrete component analysis. In *Subspace, Latent Structure and Feature Selection*, pages 1–33. Springer, 2006. 42
- A. Cakravastia and K. Takahashi*. Integrated model for supplier selection and negotiation in a make-to-order environment. *International Journal of Production Research*, 42(21):4457–4474, 2004. 52, 54
- J. M. Carroll and M. B. Rosson. Interfacing thought: cognitive aspects of human-computer interaction. chapter Paradox of the active user, pages 80–111. MIT Press, Cambridge, MA, USA, 1987. ISBN 0-262-03125-6. 44
- S. Chakrabarti, B. Dom, D. Gibson, S. R. Kumar, P. Raghavan, S. Rajagopalan, and A. Tomkins. Experiments in topic distillation. In *ACM SIGIR workshop on Hypertext Information Retrieval on the Web*. Melbourne, Australia, 1998a. 57
- S. Chakrabarti, B. Dom, P. Raghavan, S. Rajagopalan, D. Gibson, and J. Kleinberg. Automatic resource compilation by analyzing hyperlink structure and associated text. *Computer Networks and ISDN Systems*, 30(1):65–74, 1998b. 57
- C.-T. Chen, C.-T. Lin, and S.-F. Huang. A fuzzy approach for supplier evaluation and selection in supply chain management. *International Journal of Production Economics*, 102(2):289–301, 2006. 52, 53
- S.-H. Chen, H.-T. Lee, and Y.-F. Wu. Applying anp approach to partner

BIBLIOGRAPHY

- selection for strategic alliance. *Management Decision*, 46(3):449–465, 2008. 52, 144
- P.-A. Chirita, C. S. Firan, and W. Nejdl. Summarizing local context to personalize global web search. In *Proceedings of the 15th ACM international conference on Information and knowledge management, CIKM '06*, pages 287–296, New York, NY, USA, 2006. ACM. ISBN 1-59593-433-2. 43, 44, 45, 48, 57, 59
- K. Choy and W. Lee. A generic supplier management tool for outsourcing manufacturing. *Supply Chain Management: An International Journal*, 8(2): 140–154, 2003. 52, 53, 144
- M. D. V. Christopher, G. Shlomo, and T. Andrew. Document clustering evaluation: Divergence from a random baseline. *CoRR*, abs/1208.5654, 2012. 104
- J. E. Cohen and U. G. Rothblum. Nonnegative ranks, decompositions, and factorizations of nonnegative matrices. *Linear Algebra and its Applications*, 190:149–168, 1993. 42
- Y. Crama, R. Pascual J, and A. Torres. Optimal procurement decisions in the presence of total quantity discounts and alternative product recipes. *European Journal of Operational Research*, 159(2):364–378, 2004. 53
- M. T. Dacin, M. A. Hitt, and E. Levitas. Selecting partners for successful

BIBLIOGRAPHY

- international alliances: examination of us and korean firms. *Journal of world business*, 32(1):3–16, 1997. 144
- A. S. Das, M. Datar, A. Garg, and S. Rajaram. Google news personalization: scalable online collaborative filtering. In *Proceedings of the 16th international conference on World Wide Web, WWW '07*, pages 271–280, New York, NY, USA, 2007. ACM. ISBN 978-1-59593-654-7. 44, 57
- V. David, C. Iván, and M. J. Joemon. Personalizing web search with folksonomy-based user and document profiles. In *ECIR*, pages 420–431, 2010. 12, 45, 48, 50, 125, 128, 129
- G. W. Dickson. An analysis of vendor selection systems and decisions. *Journal of purchasing*, 2(1):5–17, 1966. 53, 54, 141
- L. Ding, T. Finin, A. Joshi, R. Pan, R. S. Cost, Y. Peng, P. Reddivari, V. Doshi, and J. Sachs. Swoogle: a search and metadata engine for the semantic web. In *Proceedings of the thirteenth ACM international conference on Information and knowledge management*, pages 652–659. ACM, 2004. 83
- B. E. Dom. An information-theoretic external cluster-validity measure. In *Proceedings of the Eighteenth conference on Uncertainty in artificial intelligence, UAI'02*, pages 137–145, San Francisco, CA, USA, 2002. Morgan Kaufmann Publishers Inc. ISBN 1-55860-897-4. 104
- Z. Dou, R. Song, and J.-R. Wen. A large-scale evaluation and analysis of personalized search strategies. In *Proceedings of the 16th international con-*

BIBLIOGRAPHY

- ference on World Wide Web*, WWW '07, pages 581–590, New York, NY, USA, 2007. ACM. ISBN 978-1-59593-654-7. 48
- P. Drineas, R. Kannan, and M. W. Mahoney. Fast monte carlo algorithms for matrices ii: Computing a low-rank approximation to a matrix. *SIAM Journal on Computing*, 36(1):158–183, 2006a. 41
- P. Drineas, R. Kannan, and M. W. Mahoney. Fast monte carlo algorithms for matrices iii: Computing a compressed approximate matrix decomposition. *SIAM Journal on Computing*, 36(1):184–206, 2006b. 41
- P. Drineas, M. W. Mahoney, and S. Muthukrishnan. Subspace sampling and relative-error matrix approximation: Column-based methods. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, pages 316–326. Springer, 2006c. 41
- P. Drineas, M. W. Mahoney, and S. Muthukrishnan. Subspace sampling and relative-error matrix approximation: Column-row-based methods. In *Algorithms-ESA 2006*, pages 304–314. Springer, 2006d. 41
- P. Drineas, M. W. Mahoney, and S. Muthukrishnan. Relative-error cur matrix decompositions. *SIAM Journal on Matrix Analysis and Applications*, 30(2): 844–881, 2008. 41
- R. Dulmin and V. Mininno. Supplier selection using a multi-criteria decision aid method. *Journal of Purchasing and Supply Management*, 9(4):177–187, 2003. 52, 53

BIBLIOGRAPHY

- J. H. Dyer. *Collaborative advantage: Winning through extended enterprise supplier networks*. Oxford University Press New York, 2000. 144
- M. V. Ellen. The trec-8 question answering track report. In *In Proceedings of TREC-8*, pages 77–82, 1999. 108
- D. C. Engelbart. Augmenting Human Intellect: A Conceptual Framework. Air Force Office of Scientific Research, AFOSR-3233, www.bootstrap.org/augdocs/friedewald030402/augmentinghumanintellect/ahi62index.html, 1962. 37
- A. Eugene, B. Eric, D. Susan, and R. Robert. Learning user interaction models for predicting web search result preferences. In *SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 3–10. ACM, 2006. 43
- A. Fabian, H. Nicola, H. Eelco, and K. Daniel. Interweaving public user profiles on the web. In *UMAP*, pages 16–27. Springer, 2010. 45
- D. Fensel. Ontologies: A silver bullet for knowledge management and electronic-commerce (2000). *Berlin: Spring-Verlag*. 143
- P. Ferragina and A. Gulli. A personalized search engine based on web-snippet hierarchical clustering. In *Special interest tracks and posters of the 14th international conference on World Wide Web, WWW '05*, pages 801–810, New York, NY, USA, 2005. ACM. ISBN 1-59593-051-5. 48, 49, 57, 59

BIBLIOGRAPHY

- T. Fred. *From Counterculture to Cyberculture: Stewart Brand, the Whole Earth Network, and the Rise of Digital Utopianism*. The University of Chicago Press, Chicago, Ill., 2006. ISBN 0-226-81741-5. 37
- S. Gauch, J. Chaffee, and A. Pretschner. Ontology based personalized search and browsing. *Web Intelli. and Agent Sys.*, 1:219-234, December 2003. ISSN 1570-1263. 45, 48, 57
- H. G. Gene and F. V. L. Charles. *Matrix Computations*. 1996. 40, 41
- R. E. Giachetti. A framework to review the information integration of the enterprise. *International Journal of Production Research*, 42(6):1147-1166, 2004. 135
- J. C. Gower and G. Ross. Minimum spanning trees and single linkage cluster analysis. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 18(1):54-64, 1969. 99
- C. Hans. Supporting partner identification for virtual organisations in manufacturing. *Journal of Manufacturing Technology Management*, 19(4):497-513, 2008. 144
- H. M. Hassan. Clustering web images using association rules, interestingness measures, and hypergraph partitions. In *In: ICWE 06: Proceedings of the 6th international conference on Web engineering*, pages 48-55. ACM Press, 2006. 104

BIBLIOGRAPHY

- T. H. Haveliwala. Topic-sensitive pagerank. In *Proceedings of the 11th international conference on World Wide Web*, pages 517–526. ACM, 2002. 57
- M. Heckner, M. Heilemann, and C. Wolff. Personal information management vs. resource sharing: Towards a model of information behavior in social tagging systems. In *ICWSM, 2009*. 39
- M. Hepp. Goodrelations: An ontology for describing products and services offers on the web. In *Knowledge Engineering: Practice and Patterns*, pages 329–346. Springer, 2008. 140
- T. Hofmann. Probabilistic latent semantic indexing. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 50–57. ACM, 1999a. 42
- T. Hofmann. Probabilistic latent semantic analysis. In *Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence*, pages 289–296. Morgan Kaufmann Publishers Inc., 1999b. 42
- T. Hofmann. Unsupervised learning by probabilistic latent semantic analysis. *Machine learning*, 42(1-2):177–196, 2001. 42
- A. Hotho, R. Jäschke, C. Schmitz, and G. Stumme. Emergent semantics in bibsonomy. *GI Jahrestagung (2)*, 94:305–312, 2006. 38
- A. Ioannis, A. Konstantinos, and M. J. Joemon. A comparison of general vs personalised affective models for the prediction of topical relevance. In *SIGIR*, pages 371–378, 2010. 10

BIBLIOGRAPHY

- P. K. Jagersma and D. M. van Gorp. Still searching for the pot of gold: doing business in todays china. *Journal of Business Strategy*, 24(5):27–35, 2003. 144
- K. Järvelin and J. Kekäläinen. Cumulated gain-based evaluation of ir techniques. *ACM Trans. Inf. Syst.*, 20:422–446, October 2002. ISSN 1046-8188. 73, 74
- H. Jiawei and K. Micheline. *Data Mining: Concepts and techniques*. 2001. 40
- N. J. Kaplan and J. Hurd. Realizing the promise of partnerships. *Journal of Business Strategy*, 23(3):38–42, 2002. 144
- L. Kaufman and P. J. Rousseeuw. *Finding Groups in Data: An Introduction to Cluster Analysis*. John Wiley, 1990. ISBN 1-58133-109-7. 106
- J. Kay and R. Kummerfeld. An individualised course for the c programming language. In *Proceedings of Second International WWW Conference*, pages 17–20, 1994. 7
- D. Kelly and J. Teevan. Implicit feedback for inferring user preference: a bibliography. *SIGIR Forum*, 37:18–28, September 2003. ISSN 0163-5840. 43
- S. Koshman, A. Spink, and B. J. Jansen. Web searching on the vivisimo search engine. *J. Am. Soc. Inf. Sci. Technol.*, 57:1875–1887, December 2006. ISSN 1532-2882. 48, 57

BIBLIOGRAPHY

- R. Kraft and J. Zien. Mining anchor text for query refinement. In *Proceedings of the 13th international conference on World Wide Web*, pages 666–674. ACM, 2004. 62
- A. Krüger, J. Baus, D. Heckmann, M. Kruppa, and R. Wasinger. Adaptive mobile guides. In *The adaptive web*, pages 521–549. Springer, 2007. 8
- H. Kumar and S. Kang. Another face of search engine: Web search api’s. In *Proceedings of the 21st international conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems: New Frontiers in Applied Artificial Intelligence, IEA/AIE ’08*, pages 311–320, Berlin, Heidelberg, 2008. Springer-Verlag. ISBN 978-3-540-69045-0. 48
- H. Kumar and H.-G. Kim. Semantically enriched user interest profile built from users’ tweets. In *ICADL*, pages 333–337, 2012. 39, 45
- H. Kumar and H.-G. K. Kim. Using folksonomies for building user interest profile. In *UMAP*, pages 438–441, 2011. 39, 45
- D. D. Lee and H. S. Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–791, 1999. 42
- J. S. Lee, N. Lee, C. H. Noh, and Y. Han. Ontology-based job scheduling using mobile agent technology in grid computing. *Information: An International Interdisciplinary Journal*, 13(5):1639–1651, 2010. 54
- K. Lee, H. Kim, H. Shin, and H.-J. Kim. Tag sense disambiguation for clarifying the vocabulary of social tags. In *Computational Science and Engineering*,

BIBLIOGRAPHY

2009. *CSE'09. International Conference on*, volume 4, pages 729–734. IEEE, 2009. 39
- F. Lemke, K. Goffin, and M. Szwejczeński. Investigating the meaning of supplier-manufacturer partnerships: an exploratory study. *International Journal of Physical Distribution & Logistics Management*, 33(1):12–35, 2003. 144
- L. Li, B. Wu, and Y. Yang. An ontology-oriented approach for virtual enterprises. In *Advanced Web Technologies and Applications*, pages 834–843. Springer, 2004a. 52, 54
- M. Li, X. Chen, X. Li, B. Ma, and P. M. Vitányi. The similarity metric. *Information Theory, IEEE Transactions on*, 50(12):3250–3264, 2004b. 65, 67
- Y. Li, B. Huang, W. Liu, H. Gou, and C. Wu. Ontology based decision support system for partner selection of virtual enterprises. In *Systems, Man, and Cybernetics, 2001 IEEE International Conference on*, volume 3, pages 2041–2045. IEEE, 2001. 52, 54
- J. Lin, X. Xu, and D. Xu. Strategic supplier selection: A domain driven data mining methodology. *Information*, 13(4):1449–1465, 2010. 52
- F.-H. F. Liu and H. L. Hai. The voting analytic hierarchy process method for selecting supplier. *International Journal of Production Economics*, 97(3): 308–317, 2005. 52, 144

BIBLIOGRAPHY

- J. López, E. Millán, J. Pérez-de-la Cruz, and F. Triguero. Ilesa: a web-based intelligent learning environment for the simplex algorithm. In *Proc. of CALISCE*, volume 98, pages 399–406, 1998. 4
- Z. Ma, G. Pant, and O. R. L. Sheng. Interest-based personalized search. *ACM Trans. Inf. Syst.*, 25, February 2007. ISSN 1046-8188. 43
- B. Maheshwari, V. Kumar, and U. Kumar. Optimizing success in supply chain partnerships. *Journal of Enterprise Information Management*, 19(3):277–291, 2006. 144
- C. D. Manning, P. Raghavan, and H. Schtze. *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA, 2008. ISBN 0521865719, 9780521865715. 73
- D. McArthur, C. Stasz, J. Hotta, O. Peter, and C. Burdorf. Skill-oriented task sequencing in an intelligent tutor for basic algebra. *Instructional Science*, 17(4):281–307, 1988. 6
- O. A. McBryan. Genvl and www: Tools for taming the web. In *Proceedings of the first international world wide web conference*, volume 341. Geneva, 1994. 62
- G. I. McCalla, R. B. Bunt, and J. J. Harms. The design of the scent automated advisor. *Computational Intelligence*, 2(1):76–92, 1986. 6
- H. Min. International supplier selection:: A multi-attribute utility approach.

BIBLIOGRAPHY

- International Journal of Physical Distribution & Logistics Management*, 24 (5):24–33, 1994. 52, 53
- M. G. Noll and C. Meinel. Web search personalization via social bookmarking and tagging. In *Proceedings of the 6th international The semantic web and 2nd Asian conference on Asian semantic web conference, ISWC'07/ASWC'07*, pages 367–380, Berlin, Heidelberg, 2007. Springer-Verlag. ISBN 3-540-76297-3, 978-3-540-76297-3. xii, 12, 13, 45, 46, 47, 48, 49, 85, 128, 166
- Y. Okazaki, K. Watanabe, and H. Kondo. An implementation of the www based its for guiding differential calculations. In *Proc. of Workshop " Intelligent Educational Systems on the World Wide Web" at 8th World Conference on Artificial Intelligence in Education*, pages 18–25, 1997. 4
- P. Paatero and U. Tapper. Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values. *Environmetrics*, 5(2):111–126, 1994. 42
- C. H. Papadimitriou, H. Tamaki, P. Raghavan, and S. Vempala. Latent semantic indexing: A probabilistic analysis. In *Proceedings of the seventeenth ACM SIGACT-SIGMOD-SIGART symposium on Principles of database systems*, pages 159–168. ACM, 1998. 42
- E. Pariser. *The Filter Bubble: What the Internet Is Hiding from You*. Penguin Press, New York, NY, USA, 2011. ISBN 978-1-59420-300-8. 168

BIBLIOGRAPHY

- G.-P. A. Pérez, A. Zubiaga, V. Fresno, and R. Martínez. Reorganizing clouds: A study on tag clustering and evaluation. *Expert Syst. Appl.*, 39(10):9483–9493, Aug. 2012. ISSN 0957-4174. 104
- S. A. Petersen and M. Divitini. Using agents to support the selection of virtual enterprise teams. *AOIS@ AAMAS*, 59, 2002. 52, 54
- X. H. Pham and J. J. Jung. Exploiting semantic template for message summarization for mobile devices. *Information: An International Interdisciplinary Journal*, 13(4):1467–1474, 2010. 54
- T. A. Phelps and R. Wilensky. Robust hyperlinks cost just five words each. 2000. 68
- A. B. Pidduck. Issues in supplier partner selection. *Journal of Enterprise Information Management*, 19(3):262–276, 2006. 144
- J. Pitkow, H. Schütze, T. Cass, R. Cooley, D. Turnbull, A. Edmonds, E. Adar, and T. Breuel. Personalized search. *Commun. ACM*, 45(9):50–55, Sept. 2002. ISSN 0001-0782. 48
- S. Rennick-Egglestone, A. Whitbrook, C. Leygue, J. Greensmith, B. Walker, S. Benford, H. Schnädelbach, S. Reeves, J. Marshall, D. Kirk, et al. Personalizing the theme park: psychometric profiling and physiological monitoring. In *User Modeling, Adaption and Personalization*, pages 281–292. Springer, 2011. 5

BIBLIOGRAPHY

- R. Riding and S. Rayner. *Cognitive styles and learning strategies: Understanding style differences in learning and behaviour*. D. Fulton Publishers, 1998. 7
- D. E. Rose and D. Levinson. Understanding user goals in web search. In *Proceedings of the 13th international conference on World Wide Web, WWW '04*, pages 13–19, New York, NY, USA, 2004. ACM. ISBN 1-58113-844-X. 168
- G. Rossi, D. Schwabe, and R. Guimarães. Designing personalized web applications. In *Proceedings of the 10th international conference on World Wide Web*, pages 275–284. ACM, 2001. 5
- P. J. Rousseeuw. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20(1):53–65, Nov. 1987. ISSN 03770427. 105
- G. Salton. The smart retrieval system experiments in automatic document processing. 1971. 56
- G. Salton, A. Wong, and C.-S. Yang. A vector space model for automatic indexing. *Commun. ACM*, 18(11):613–620, 1975. 102, 155
- J. Schmitz and K. Platts. Supplier logistics performance measurement: indications from a study in the automotive industry. *International Journal of Production Economics*, 89(2):231–243, 2004. 53

BIBLIOGRAPHY

- P. Schmitz. Inducing ontology from flickr tags. In *Collaborative Web Tagging Workshop at WWW2006, Edinburgh, Scotland*, volume 50, 2006. 39
- D. Scott, T. D. Susan, W. F. George, K. L. Thomas, and H. Richard. Indexing by latent semantic analysis. *JOURNAL OF THE AMERICAN SOCIETY FOR INFORMATION SCIENCE*, 41(6):391–407, 1990. 91, 98
- S. Shekhar. Benchmarking knowledge gaps through role simulations for assessing outsourcing viability. *Benchmarking: An International Journal*, 15(3): 225–241, 2008. 144
- X. Shen, B. Tan, and C. Zhai. Context-sensitive information retrieval using implicit feedback. In *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval, SIGIR '05*, pages 43–50, New York, NY, USA, 2005. ACM. ISBN 1-59593-034-5. 43, 44
- B. Sigurbjörnsson and Z. R. Van. Flickr tag recommendation based on collective knowledge. In *Proceedings of the 17th international conference on World Wide Web*, pages 327–336. ACM, 2008. 39
- A. P. Singh and G. J. Gordon. A unified view of matrix factorization models. In *Machine Learning and Knowledge Discovery in Databases*, pages 358–373. Springer, 2008. 42
- M. Speretta and S. Gauch. Personalized search based on user search histories. In *Proceedings of the 2005 IEEE/WIC/ACM International Conference on*

BIBLIOGRAPHY

- Web Intelligence*, WI '05, pages 622–628, Washington, DC, USA, 2005a. IEEE Computer Society. ISBN 0-7695-2415-X. 45, 48
- M. Speretta and S. 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, 2005b. 57
- Z. Sui and Q. Zhao. To extract ontology attribute value automatically based on www. *Information: An International Interdisciplinary Journal*, 12(2), 2009. 54
- J.-T. Sun, H.-J. Zeng, H. Liu, Y. Lu, and Z. Chen. Cubesvd: a novel approach to personalized web search. In *Proceedings of the 14th international conference on World Wide Web*, pages 382–390. ACM, 2005. 57
- Q. Sun, J. Ji, and W. Xu. A new approach for vendor evaluation and selection based on maximizing deviation multiple attribute decision. *Information: An International Interdisciplinary Journal*, 12(1):13–19, 2009. 52, 53
- P.-N. Tan, M. Steinbach, and V. Kumar. *Introduction to Data Mining, (First Edition)*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2005. ISBN 0321321367. 113
- F. Tarpin-Bernard and H. Habieb-Mammar. Modeling elementary cognitive abilities for adaptive hypermedia presentation. *User Modeling and User-Adapted Interaction*, 15(5):459–495, 2005. 8

BIBLIOGRAPHY

- J. Teevan, S. Dumais, and E. Horvitz. Personalizing search via automated analysis of interests and activities. In *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '05, 2005. ISBN 1-59593-034-5. 43, 44, 57, 59
- J. Teevan, S. T. Dumais, and E. Horvitz. Characterizing the value of personalizing search. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '07, pages 757–758, New York, NY, USA, 2007. ACM. ISBN 978-1-59593-597-7. 43, 57
- O. Tim. What is web 2.0: Design patterns and business models for the next generation of software. 2005. doi: <http://dx.doi.org/10.1016/j.websem.2007.11.011>. URL <http://www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html>. 36
- G. Tom. Collective knowledge systems: Where the social web meets the semantic web. *Web Semantics: Science, Services and Agents on the World Wide Web*, 6(1):4 – 13, 2008. ISSN 1570-8268. doi: <http://dx.doi.org/10.1016/j.websem.2007.11.011>. URL <http://www.sciencedirect.com/science/article/pii/S1570826807000583>. 35
- V. Tsiriga and M. Virvou. Modelling the student to individualise tutoring in a web-based icall. *International Journal of Continuing Engineering Education and Life Long Learning*, 13(3):350–365, 2003. 4

BIBLIOGRAPHY

- V. Uren, P. Cimiano, J. Iria, S. Handschuh, M. Vargas-Vera, E. Motta, and F. Ciravegna. Semantic annotation for knowledge management: Requirements and a survey of the state of the art. *Web Semantics: science, services and agents on the World Wide Web*, 4(1):14–28, 2006. 83
- D. Vallet and P. Castells. Personalized diversification of search results. In *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval, SIGIR '12*, pages 841–850, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1472-5. 45
- E. Van Couvering. The history of the internet search engine: Navigational media and the traffic commodity. In *Web Search*, pages 177–206. Springer, 2008. 56
- W. T. Vander. Explaining and showing broad and narrow folksonomies. <http://www.vanderwal.net/random/entrysel.php?blog=1635>, 2005. 39
- W. T. Vander. Folksonomy. <http://www.vanderwal.net/essays/051130/folksonomy.pdf>, Feb 2007. 39
- J. Vassileva. An architecture and methodology for creating a domain-independent, plan-based intelligent tutoring system. *Programmed Learning and Educational Technology*, 27(4):386–397, 1990. 6
- J. Vassileva. A task-centered approach for user modeling in a hypermedia office documentation system. *User modeling and user-adapted interaction*, 6(2-3): 185–223, 1996. 7

BIBLIOGRAPHY

- L. Wang and P. Kess. Partnering motives and partner selection: Case studies of finnish distributor relationships in china. *International Journal of Physical Distribution & Logistics Management*, 36(6):466–478, 2006. 144
- Q. Wang and H. Jin. Exploring online social activities for adaptive search personalization. In *Proceedings of the 19th ACM international conference on Information and knowledge management*, CIKM, pages 999–1008, New York, NY, USA, 2010. ACM. ISBN 978-1-4503-0099-5. 18, 48
- C. A. Weber, J. R. Current, and W. Benton. Vendor selection criteria and methods. *European journal of operational research*, 50(1):2–18, 1991. 54, 141
- S. Xu, S. Bao, B. Fei, Z. Su, and Y. Yu. Exploring folksonomy for personalized search. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '08, 2008. ISBN 978-1-60558-164-4. 12, 45, 46, 47, 49, 50, 86, 125, 128, 166

Appendices

.1 Pairs of Query and target URL

List of self-evident query and target URL

Table 1: List of Self-evident query and target URL pairs

Query	Target URL	Query	Target URL
Puzzle	zigzone.com	Math	mathlesson.com
Medicine	jmir.org	Estoer	en.wikipedia.org/ wiki/George_Gurdjieff
Hostel	en.wikibooks.org/ wiki/LaTeX/Tables	Radio	planetradiocity.com/ internetradio/index.php
amazon	amazon.com	bollywood	bollywoodhungama.com/ trade/releasedates/ index.html
Basketball	nba.com	Pbs	www.pbs.org
Islam	islamtoday.com	Boardgame	boardgamers.org
Columbia	columbia.edu	Redcross	redcross.org
Imdb	imdb.com	Thinkquest	library.thinkquest.org
Overstock	overstock.com	Gap	gap.com
Walmart	walmart.com	Ebay	cgi.ebay.com
Wikipedia	en.wikipedia.org	Citibank	citibank.com
Kraft	kraftfoods.com	Mapquest	mapquest.com
Dictionary	dictionary.com	Costco	costco.com

Continued on next page

.1 Pairs of Query and target URL

Table 1 – *Continued from previous page*

Query	Target URL	Query	Target URL
Fbi	fbi.gov	Starbucks	starbucks.com
Mtv	mtv.com	Cisco	cisco.com
Marriott	marriott.com	Weather	weather.com
Hasbro	hasbro.com	Metlife	metlife.com
Bbc	bbc.co.uk	Playboy	playboy.com
Businessweek	businessweek.com	Washingtonpost	washingtonpost.com
Whitehouse	whitehouse.gov	Time	timeanddate.com
Carter	carters.com	Skype	skype.com
Microsoft	microsoft.com	Flickr	flickr.com
Oldnavy	oldnavy.com	Patent	freepatentsonline.com
Sports	qcbaseball.com	Princeton	princeton.edu
e-health	electronic- health.org/	jigsaw puzzle	jigzone.com

List of vague query and target URL

Table 2: List of vague query and target URL pairs

Query	Target URL	Query	Target URL
Magazine	automobilemagazine.com	Planet	solarspace.co.uk

Continued on next page

.1 Pairs of Query and target URL

Table 2 – *Continued from previous page*

Query	Target URL	Query	Target URL
Auction	ragoarts.com	Worksheet	abcteach.com
Latex	betweentheshets.co.uk	Business	alibaba.com
History	onwar.com	latex	en.wikibooks.org/ wiki/LaTeX/Mathematics
Telephone	skype.com	Keynote	apple.com/ iwork/keynote/
Apple	kronenberg.org	Electronics	radioshack.com
divorce	divorcenet.com	Travel	chowbaby.com
Legal	womenslaw.com	Manufacture	tradekey.com
Realtor	foxtons.com	Food	chinesefood.about.com
Quiz	iqtest.com	Queen	queenszoo.com
Price comparison	calibex.com	Gold	Taxfreegold.co.uk
History	bible-history.com	Music	traditionalmusic.com
Entertainment	playboy.com	Database	freepatentsonline.org
Religion	cyberhymnal.org	Bible	studylight.org
Sports	qcbaseball.com	Newspaper	alligator.org
Religion	tenets.zoroastrianism.com	Stories	skywriting.net
Music	hymnal.net	Philosophy	vbm-torah.org
Automobile	kbb.com	Pond	ponds.com
Worship	Textweek.com	Health	holisticjunctino.com

Continued on next page

.2 Examples of Expanded Queries

Table 2 – *Continued from previous page*

Query	Target URL	Query	Target URL
Assist	Natri.uky.edu	Travel	ryanair.com

.2 Examples of Expanded Queries

1. The query pond was disambiguated by the cluster [beauty, products] thus pushing the `www.ponds.com` at the top of the result set.
2. The query religion is a very good example where cluster structure plays an important role. For one user who had interest in Christianity, the query religion was rightly disambiguated with the cluster [religion, Christian, church, catholic] resulting in URL `www.cyberhymnal.org` at higher rank. For another user, the same query religion was mapped to a cluster [moshiach, judaism, jewish, mysteri, mashiach, messiah] to disambiguate the context of term religion which resulted in the URL `tenets.zoronastrianism.com` promoted to the top position.
3. Another query latex was mapped to [latex, fetish, sheet, rubber, shop, house, satin, bed] pushing up the URL `www.betweenthesheets.co.uk` at the top position and lowering the rank of URLs related to Latex document markup language.

.3 An example of svdCUIP, modSvdCUIP, tfIdfCUIP

tfIdfCUIP (d=0.09)

[[ngo], [scuba, korea, dive], [editplu, softwar, regex],
[bollywood, releas, movi, hindi], [whitespac, tab, tip, format],
[data, excel, import, csv, financi, microsoft], [fm, music, radio],
[dna, genealog, genet, scienc, technolog, biologi],
[wp, wealth, wealthi, life, busi, mexico, philanthropi, person, slim,
biographi], [log, overview, classif, datamin, queri],
[video, divx, download, legenda, subtitl, film],
[free, skype, voip, telephoni, phone],
[supermercado, carrefour, casa, onlin, compra, spanish, tienda],
[comida, food, restaurant],
[mac, osx, wine, virtual, wikipedia, window, resourc, emul, linux],
[iwork, tutori, imovi, train, gwt, appl, ilif],
[lowcost, europ, vuelo, airlin, flight, lodg, travel, vacat, hotel],
[store, preppi, cheap, deal, watch, men, wear, fashion, cloth, brand,
shop, women], [financ, theater, card, bank, creditcard, cg, samsung],
[algoritmo, poll, code, cs, binari, soa, backoff, algorithm, program,
exponentialbackoff],
[statist, decis, ahp, lean, manag, multicriteria, decisionmak, engin,

.3 An example of svdCUIP, modSvdCUIP, tflDfCUIP

projectmanag, hierarchi, analysi, process, econom, analyt],
[fourthwai, magic, spiritu, happi, learn, gurdjieff, epicurean, charact,
occult, philosophi, epicuru, esoter, osho, book],
[datetim, databas, mysql, date, creat, php, sql, exampl, function, develop],
[refer, document, latex, style, notat, packag, command, wiki, custom, tex],
[viaj, hostel, espa, airport, barcelona, spain, hostel],
[ebm, review, bmj, patient, new, cochran, socialnetwork, collabor, social,
health, commun, healthcar, medicin, medic, drug],
[openoffic], [fabul], [web2.0, semant_web, elearn, forschung, educ],
[wikibook, tabl], [float, howto, imag, figur], [firefox, extens, check],
[perform, tcpip, congest, tech, tcp, network],
[math, mathemat, verbal, teach],
[2011, confer, android:bookmark, hci, research],
[inform_scienc, inform, ci, inform-scienc, journal, li],
[chrome, webkit, tool, typographi, opensourc, typeset, browser],
[time, est, timezon, dst, convert, standard],
[matrix, librari, machin-learn, ai, java, api, algebra, machinelearn],
[load, graphic, color, comput, manual],
[entertain, kid, puzzl, interact, fun, game, jigsaw],
[informat, ehealth, internet, cfp, e-health],
[seo], [space], [paper], [export, file], [write, mactex, macosx],
[postscript, subfigur], [subscript, superscript], [shell, output],
[powerpoint, keynot, present, design],

.3 An example of svdCUIP, modSvdCUIP, tflidfCUIP

[astronomi, telrad, telescop], [cheatsheet, symbol],
[cook, restaur, vegetarian, vegan, guid],
[my.cnf, db, config, configur, backup, work, ini],
[exam, question, certif, test, scjp, mock, certification, certifica],
[babi, carter, crian, children, apparel],
[taxonomi, ux, usabl, ui, toread, ia],
[2012, lyon, public, www, www2012, via:packrati.us]]

svdCUIP(k=90, d=0.13)

[[babi, children, men, wear, fashion, cloth, brand, shop, women], [ngo],
[spiritu, happi, learn, gurdjieff, epicurean, occult, philosophi, epicuru],
[servic], [chrome, webkit, opensourc, browser],
[refer, howto, math, latex, tutori, wiki, tabl, symbol, gwt, figur, tex],
[question, certif, java, test, scjp, mock],
[float], [db, config, configur, work],
[2011, confer, android:bookmark, hci, research, cfp, e-health],
[osx, wine, virtual, window, resourc, emul, linux], [fourthwai],
[datetim], [bookmark], [cook], [statist], [magic], [algoritmo],
[mac, perform, tcpip, congest, tech, wikipedia, tcp, network],
[taxonomi, usabl, seo, ia], [preppi], [load], [my.cnf],
[wp, wealth, wealthi, life, busi, mexico, philanthropi, person,
slim, biographi], [exam], [review],
[free, skype, voip, telephoni, phone],

.3 An example of svdCUIP, modSvdCUIP, tflidfCUIP

[bmj], [store, cheap, deal, watch, dailyd, daili],
[poll, binari, soa, backoff, algorithm, program, exponentialbackoff],
[databas, mysql, date, shell, sql, function, output, develop], [ux],
[entertain, kid, puzzl, fun, game, jigsaw], [ebm, cochran, drug],
[patient, socialnetwork, social, commun], [document], [graphic, manual],
[decis, ahp, manag, decisionmak, engin, process, econom], [write],
[powerpoint, keynot, present, design],
[informat, ehealth, internet, journal, health, healthcar, medicin, medic],
[lowcost, europ, airlin, flight, travel, vacat, hotel],
[matrix, librari, api], [ui],
[2012, lyon, public, www, www2012, via:packrati.us],
[carter], [wikibook], [interact], [new], [openoffic], [tool], [fabul],
[typographi], [mactex], [macosx], [inform_scienc], [casa], [creat],
[cs], [crian], [code], [lean], [ci], [typeset], [style], [collabor],
[whitespac], [color], [notat], [php], [tab], [tip], [spanish], [charact],
[multicriteria], [vuelo], [projectmanag], [hierarchi], [imovi], [toread],
[packag], [analysi], [command], [space], [cheatsheet], [algebra], [backup],
[train], [custom], [exampl], [lodg], [certification], [paper], [esoter],
[format], [imag], [ini], [comput], [book], [astronomi, telrad, telescop],
[osho], [certifica], [analyt],[film], [apparel], [postscript, subfigur],
[editplu, softwar, regex], [scuba, korea, dive], [firefox, extens, check],
[subscript, superscript], [fm, music, radio],
[web2.0, semant_web, elearn, forschung, educ], [iwork, appl, ilif],

.3 An example of svdCUIP, modSvdCUIP, tflidfCUIP

[mathemat, verbal, teach], [viaj, hostel, espa, hostel],
[log, overview, classif, datamin, queri], [machin-learn, ai, machinelearn],
[dna, genealog, genet, scienc, technolog, biologi],
[theater, bollywood, releas, movi, cgv],
[airport, barcelona, comida, spain, food, restaurant],
[inform, inform-scienc, li],
[restaur, vegetarian, vegan, guid],
[export, file], [video, divx, download, legenda, subtitl],
[data, excel, import, csv, financi, microsoft],
[supermercado, carrefour, onlin, compra, tienda],
[time, est, timezon, dst, convert, standard],
[financ, card, bank, creditcard, samsung]]

modSvdCUIP(k=100, d=0.63)

[[ngo], [happi, learn, epicurean, philosophi, epicuru],
[patient, socialnetwork, collabor, social, commun],
[fm, music, india, radio], [matrix, api, algebra],
[bollywood, releas, movi, hindi], [editplu, softwar, regex],
[exam, question, certif, java, test, scjp, mock, certification, certifica]
[math, mathemat, verbal, teach],
[preppi, men, wear, fashion, cloth, brand, women],
[supermercado, carrefour, casa, onlin, compra, spanish, tienda],
[financ, card, bank, creditcard, samsung],

.3 An example of svdCUIP, modSvdCUIP, tflDfCUIP

[inform_scienc, inform, ci, inform-scienc, li],
[scuba, korea, dive][time, dst], [kid, game],
[viaj, hostel, espa, hostel], [float, imag, figur],
[dna, genealog, genet, technolog, biologil],
[log, overview, classif, datamin, queri],
[video, divx, download, legenda, subtitl, film],
[wp, wealth, wealthi, life, busi, mexico, philanthropi,
person, slim, biographi],
[barcelona, spain], [openoffic], [fabul], [lodg, travel, vacat],
[data, excel, import, csv, financi, microsoft],
[free, skype, voip, telephoni, phone],
[tool, opensourc], [2011, confer, android:bookmark, hci, cfp, e-health],
[mac, wikipedia],[graphic, color, manual], [iwork, imovi, train, appl, ilif]
[perform, tcpip, congest, tech, tcp, network, linux],
[servic, search_to_rss, search, bookmark, web, rss, feed, googl],
[osx, wine, virtual, window, resourc, emul],
[librari, machin-learn, ai, machinelearn, program],
[store, cheap, deal, watch, shop, dailyd, daili],
[refer, document, howto, latex, typographi, style, typeset, whitespac,
notat, tab, tip, packag, space, command, wiki, cheatsheet, custom, symbol,
format, tex],
[algoritmo, poll, code, cs, binari, soa, backoff, algorithm,
exponentialbackoff],

.3 An example of svdCUIP, modSvdCUIP, tflDfCUIP

[firefox, extens, check],
[statist, decis, ahp, lean, manag, multicriteria, decisionmak, engin,
projectmanag, hierarchi, analysi, process, econom, analyt],
[bmj, new, informat, ehealth, journal, health, healthcar, medicin, medic],
[datetim, databas, load, mysql, date, creat, php, sql, exampl,
function, comput, develop],
[wikibook, tutori, tabl],
[web2.0, semant_web, elearn, forschung, educ],
[internet],[seo],[airport],[scienc],[research],[gwt],[paper],[food],
[hotel],[book],
[est, timezon, convert, standard],
[comida, restaurant],[2012, lyon, public, www, www2012, via:packrati.us],
[ebm, review, cochrans, drug],
[my.cnf, db, config, configur, backup, work, ini],
[powerpoint, keynot, present, design],
[astronomi, telrad, telescop],[entertain, puzzl, interact, fun, jigsaw],
[postscript, subfigur],
[cook, restaur, vegetarian, vegan, guid],
[babi, carter, crian, children, apparel],
[subscript, superscript],
[export, file],[lowcost, europ, vuelo, airlin, flight],
[theater, cgv],
[fourthwai, magic, spiritu, gurdjieff, charact, occult, esoter, osho],

.3 An example of svdCUIP, modSvdCUIP, tldfCUIP

```
[chrome, webkit, browser],[write, mactex, macosx],  
[shell, output], [taxonomi, ux, usabl, ui, toread, ia]]
```

초록

개인화 검색 및 파트너십 선정을 위한 사용자 프로파일링

변화의 비밀은 당신의 에너지를 기존 산물에 대한 비난이나 비판이 아닌 새로운 것을 구축하는데 집중하는 것이다
- 소크라테스

사용자 관심사의 자동적 식별은 도전적인 과제임과 동시에 추천 시스템에 있어 필수적이며 핵심적인 기능이라 할 수 있다. 본 학위 논문에서는, 사용자의 관심사 혹은 선호도를 식별하고 표현하는 문제를 프로파일 작성으로 치환하여 접근한다. 사용자의 관심사를 자동적으로 추론하고, 추론된 관심사 내에 잠재된 의미를 추출하는 알고리즘들을 제안하며, 제안된 알고리즘들은 개인화 검색 성능의 향상에 초점을 맞추어 고안되었다. 또한, 사용자의 프로파일을 구매자와 판매자 프로파일로 구분하여 모델링하는 방법론을 소개하며, 프로파일을 구성하는 속성들은 규정화된 용어집 (Controlled vocabulary)에 정의된 용어를 차용한다.

개인화 검색 (Personalized search) 지원을 위해 가장 먼저, Anchor text를 활용하여 사용자의 관심사를 구축하는 획기적인 방법론을 제안한다. 다음으로, 폭소노미 (Folksonomy) 시스템이 축적한 데이터에 기반하여, 행렬인수분해 (Matrix factorization) 기법을 활용, 사용자 관심사 프로파일 내의 용어 간 관계 계산을 통해 사용자 프로파일을 생성하는 두 가지 방법론이 제시된다. 제시된 두 방법론의 목적은 문맥적, 의미적 그리고 문장 구성적인 관점에서 관계를 맺고 있어 상호 그룹화될 수 있는 연관 용어들 간의 숨겨진 관계를 발견하고, 이를 기반으로 하여 용어들이 사용된 문맥을 명확히 하는데 있다 할 수 있다. 요약하자면, 사용자 관심사 모델링과 개인화를 위한 프레임워크가 제안되며, 제안된 프레임워크를 개인화된 웹 검색 관점으로 그 성능 및 유효성을 검증한다. 제안된 프레임워크를 통해 구축된 사용자 관심사 프로파일은, 프로파일의 군집화 경향 및 정확도 (Clustering tendency and accuracy) 관점에서 다시 한번 분석된다. 사용자의 질의 문맥을 정확하고 명확하게 구별할 수 있는 사용자 관심사 프로파일은, 개인화 검색 성능에 지대한 영향력을 갖는다는 것을 대규모의 실험을 통해 발견할 수 있었다.

파트너십 선정 (Partnership match)을 위해, 파트너십 온톨로지 (Partnership ontology)라 일컬어지는 온톨로지를 소개한다. 본 연구에서 소개하는 파트너십 온톨로지는, 사용자

가 자신의 요구사항들을 구매자 프로파일 혹은 판매자 프로파일로 세분화하여 지정하기 위한 초석으로 사용된다. 마지막으로, 주어진 특정 구매자 프로파일과 부합하는 판매자 프로파일들에 우선순위 할당을 위해, 의미적 유사성을 계량화 할 수 있는 지표를 정의한다.

키워드: 사용자 모델링, 사용자 관심사, 사용자 선호도, 개인화 검색, 파트너십 선정

Acknowledgements

Everyone is my teacher. Some I seek. Some I subconsciously attract. Often I learn simply by observing others. Some may be completely unaware that I'm learning from them, yet I feel deeply in gratitude. - Eric Allen

First and Foremost, I am deeply and sincerely grateful to my supervisor, Professor Hong-Gee Kim, for his continuous and instructive guidance. I thank him for his patience and encouragement that carried me on through difficult times, and also for his insights and suggestions that helped to shape my research skills and critical thinking. As an advisor, he taught me practices and skills that I will use in my future career. As a mentor, he taught me how to shape ideas into proper research and how to manage research projects from its inception to final stage. He has been a great support all along the path to PhD by supporting my teaching activities, team mentoring, proposal writing, and most important gave me a lot of freedom to develop ideas.

I am glad that I had the privilege to do my Ph.D. at Biomed-

cal Knowledge Engineering (BIKE) Lab, Seoul National University. For that also, I would like to thank Prof. Hong-Gee Kim and Prof. Myoung-Hee Kim for both supporting my Ph.D. work and establishing this unique, creative, international research environment. One of the best thing about working in this lab is that I got to work with some of the best researchers. A very conducive research environment in which collaboration with great, talented colleagues (postdocs, researchers, MS students, Ph.D. students, administrative staff) become a wonderful experience that impacted both my professional life and personal life. I enjoyed the various workshops, seminars, research meetings, project meetings, and I can proudly say that I was a part of the Biomedical Knowledge Engineering Lab (BIKE).

I would like to thank the members of my committee for their careful examination and constructive advice: Professor Hyoung-Joo Kim, Professor Sang-goo Lee, Professor Hong-Gee Kim, Professor MyoungHee Kim, Professor Im Dong-Hyuk. I sincerely thank all my fellow researchers, specially, Eung-Hee Kim, Hyun NamGoong, Seong-Jae Song, and Seong-In Lee for always being helpful over the years and making my stay in Korea a wonderful experience.

Further, I want to thank all BIKE Lab colleagues for being such a

great team and making my stay in Korea such a great experience. Last but not least, I am very grateful to my dearest son, Rik Kumar, my parents and my wife for always being there when I needed them most, and for supporting me continuously throughout these years.



저작자표시-비영리-동일조건변경허락 2.0 대한민국

이용자는 아래의 조건을 따르는 경우에 한하여 자유롭게

- 이 저작물을 복제, 배포, 전송, 전시, 공연 및 방송할 수 있습니다.
- 이차적 저작물을 작성할 수 있습니다.

다음과 같은 조건을 따라야 합니다:



저작자표시. 귀하는 원저작자를 표시하여야 합니다.



비영리. 귀하는 이 저작물을 영리 목적으로 이용할 수 없습니다.



동일조건변경허락. 귀하가 이 저작물을 개작, 변형 또는 가공했을 경우에는, 이 저작물과 동일한 이용허락조건하에서만 배포할 수 있습니다.

- 귀하는, 이 저작물의 재이용이나 배포의 경우, 이 저작물에 적용된 이용허락조건을 명확하게 나타내어야 합니다.
- 저작권자로부터 별도의 허가를 받으면 이러한 조건들은 적용되지 않습니다.

저작권법에 따른 이용자의 권리는 위의 내용에 의하여 영향을 받지 않습니다.

이것은 [이용허락규약\(Legal Code\)](#)을 이해하기 쉽게 요약한 것입니다.

[Disclaimer](#)

공학박사 학위논문

User Profiling for Personalized
Search & Partnership Match

개인화 검색 및 파트너십 선정을
위한 사용자 프로파일링

2014 년 2 월

서울대학교 대학원

치의과학과 의료경영과정보학 전공

Harshit Kumar

User Profiling for Personalized Search & Partnership Match

개인화 검색 및 파트너십 선정을 위한 사용자
프로파일링

지도교수 김 홍 기

이 논문을 공학박사 학위논문으로 제출함
2013 년 12 월

서울대학교 대학원
치의과학과 의료경영과정정보학 전공
Harshit Kumar

Harshit Kumar의 박사 학위논문을 인준함
2014 년 2 월

위 원 장 _____ 김 형 주 (인)

부위원장 _____ 김 홍 기 (인)

위 원 _____ 이 상 구 (인)

위 원 _____ 김 명 기 (인)

위 원 _____ 임 동 혁 (인)

User Profiling for Personalized Search & Partnership Match

Adviser Hong-Gee Kim

Submitting a doctoral thesis of Computer
Science and Engineering
December 2013

Graduate School of Seoul National University
Department of Dental Science
Harshit Kumar

Confirming the doctoral thesis written by
Harshit Kumar
February 2014

Chair Hyung-Joo Kim (인)

Vice Chair Hon-Gee Kim (인)

Examiner Sang-Goo Lee (인)

Examiner Myeng-Ki Kim (인)

Examiner Dong-Hyuk Im (인)

Abstract

User Profiling for Personalized Search & Partnership Match

The secret of change is to focus all of your energy not on fighting the old, but on building the new. - Socrates

The automatic identification of user intention is an important but highly challenging research problem whose solution can greatly benefit information systems. In this thesis, I look at the problem of identifying sources of user interests, extracting latent semantics from it, and modelling it as a user profile. I present algorithms that automatically infer user interests and extract hidden semantics from it, specifically aimed at improving personalized search. I also present a methodology to model user profile as a buyer profile or a seller profile, where the attributes of the profile are populated from a controlled vocabulary. The buyer profiles and seller profiles are used in partnership match.

In the domain of personalized search, first, a novel method to construct a profile of user interests is proposed which is based on mining anchor text. Second, two methods are proposed to build a user profile that gather terms from a folksonomy system where matrix factorization technique is explored to discover hidden relationship between them. The objective of the methods is to discover latent relationship between terms such that contextually, semantically, and syntactically related terms could be grouped together, thus disambiguating the context of term usage. The profile of user interests is also analysed to judge its clustering tendency and clustering accuracy. Extensive evaluation indicates that a profile of user interests, that can correctly or precisely disambiguate the context of user query, has a significant impact on the personalized search quality. In the domain of partnership match, an ontology termed as partnership ontology is proposed. The attributes or concepts, in the partnership ontology, are features representing context of work. It is used by users to lay down their requirements as buyer profiles or seller profiles. A semantic similarity measure is defined to compute a ranked list of matching seller profiles for a given buyer profile.

Keywords : User Modelling, User Interests, User Preferences, Personalized Search, Partnership Match.

Student ID: 2010-31376

Contents

List of Figures	vii
List of Tables	xii
1 Introduction	1
1.1 User Profiling for Personalized Search	9
1.1.1 Motivation	10
1.1.2 Research Problems	11
1.2 User Profiling for Partnership Match	18
1.2.1 Motivation	19
1.2.2 Research Problems	24
1.3 Contributions	25
1.4 System Architecture - Personalized Search	29

1.5	System Architecture - Partnership Match	31
1.6	Organization of this Dissertation	32
2	Background	35
2.1	Introduction to Social Web	35
2.2	Matrix Decomposition Methods	40
2.3	User Interest Profile For Personalized Web Search - Non Folksonomy based	43
2.4	User Interest Profile for Personalized Web Search - Folksonomy based	45
2.5	Personalized Search	47
2.6	Partnership Match	52
3	Mining anchor text for building User Interest Profile: A non-folksonomy based personalized search	56
3.1	Exclusively Yours'	59
3.1.1	Infer User Interests	61
3.1.2	Weight Computation	64
3.1.3	Query Expansion	67
3.2	Exclusively Yours' Algorithm	68
3.3	Experiments	71

3.3.1	DataSet	72
3.3.2	Evaluation Metrics	73
3.3.3	User Profile Efficacy	74
3.3.4	Personalized vs. Non-Personalized Results .	76
3.4	Conclusions	80
4 Matrix factorization for building Clustered User Interest Profile: A folksonomy based personalized search 82		
4.1	Aggregating tags from user search history	86
4.2	Latent Semantics in UIP	90
4.2.1	Computing the tag-tag Similarity matrix . .	90
4.2.2	Tag Clustering to generate <i>svdCUIP</i> and <i>modSvdCUIP</i>	98
4.3	Personalized Search	101
4.4	Experimental Evaluation	103
4.4.1	Data Set and Experiment Methodology . . .	103
4.4.1.1	Custom Data Set and Evaluation Metrics	103
4.4.1.2	AOL Query Data Set and Evaluation Metrics	107

4.4.1.3	Experiment set up to estimate the value of k and d	107
4.4.1.4	Experiment set up to compare the proposed approaches with other approaches	109
4.4.2	Experiment Results	111
4.4.2.1	Clustering Tendency	111
4.4.2.2	Determining the value for dimension parameter, k , for the Custom Data Set	113
4.4.2.3	Determining the value of distinctness parameter, d , for the Custom data set	115
4.4.2.4	CUIP visualization	117
4.4.2.5	Determining the value of the dimension reduction parameter k for the AOL data set	119
4.4.2.6	Determining the value of distinctness parameter, d , for the AOL data set	120

4.4.2.7	Time to generate <i>svdCUIP</i> and <i>modSvd-</i> <i>CUIP</i>	122
4.4.2.8	Comparison of the <i>svdCUIP</i> , <i>modSvd-</i> <i>CUIP</i> , and <i>tfIdfCUIP</i> for different classes of queries	123
4.4.2.9	Comparing all five methods - Im- provement	124
4.4.3	Discussion	126
5	User Profiling for Partnership Match	133
5.1	Supplier Selection	137
5.2	Criteria for Partnership Establishment	140
5.3	Partnership Ontology	143
5.4	Case Study	147
5.4.1	Buyer Profile and Seller Profile	153
5.4.2	Semantic Similarity Measure	155
5.5	Discussion	160
5.6	Conclusions	162
6	Conclusion	164
6.1	Future Work	167

CONTENTS

6.1.1	Degree of Personalization	167
6.1.2	Filter Bubble	168
6.1.3	IPR issues in Partnership Match	169
	Bibliography	170
	Appendices	193
.1	Pairs of Query and target URL	194
.2	Examples of Expanded Queries	197
.3	An example of svdCUIP, modSvdCUIP, tfIdfCUIP	198

List of Figures

1.1	User Profiling features for various classes of Web Systems	2
1.2	User Profiling for Personalized Search and Partnership Match	9
1.3	Percentage of partnerships that are successful, partial successful, and failures	20
1.4	Reasons that cause failure of partnership	21
1.5	Percentage of companies who has formed joint ventures with other companies.	22
1.6	Key benefits of partnering.	23
1.7	A system architecture for building a <i>CUIP</i> and its application to personalized search	29

LIST OF FIGURES

1.8	An archetype for Partnership match, showing the flow of processes	31
3.1	System Architecture of Exclusively Your's	60
3.2	Set U represents URLs returned by a search engine and set V represents URLs clicked or downloaded by the user. On the right (b), URLs $h_1, h_2, h_3, \dots, h_n$ are hub URLs for URL V_i	61
3.3	A Snapshot of Exclusively Yours' user interface	69
3.4	(a) Display URLs, snippet and title (b) extracts anchor text and its surrounding text from hub URLs.	71
3.5	Efficacy of UIP constructed using different methods	76
3.6	(a) Cumulative Gain (CG) Curve for an individual user query (b) Discounted Cumulative Gain (DCG) for an individual user query.	77
3.7	Average Discounted Cumulative Gain (DCG) Curve and (b) Average Cumulative Gain (CG)	78
3.8	(a) Average Rank vs. each department (b) Average Rank vs. Search Engine	79
4.1	System Architecture of CUIP based Personalized Search	86

LIST OF FIGURES

4.2	Dendrogram visualization for similarity matrix <i>modSim</i>	100
4.3	Automatic Evaluation Methodology	108
4.4	Number of Clusters vs. average Silhouette Coefficient plot for <i>svdCUIP</i> and <i>modSvdCUIP</i>	112
4.5	A comparison of different value combinations of <i>k</i> and <i>d</i> Vs. average Silhouette Coefficient for <i>svdCUIP</i> average linkage	114
4.6	A comparison of different value combinations of <i>k</i> and <i>d</i> vs average Silhouette Coefficient for <i>modSvdCUIP</i> average linkage	115
4.7	A comparison of different value combinations <i>k</i> and <i>d</i> vs <i>AverageFscores</i> for the <i>modSvdCUIP</i> (when <i>k</i> =30,40) and the <i>svdCUIP</i> (when <i>k</i> =90,100) for average linkage.	117
4.8	Estimating the values of dimension parameter for <i>svdCUIP</i> and <i>modSvdCUIP</i> using the Improvement as an evaluation metric	119
4.9	Estimating the values of distinctness parameter for <i>tJdfCUIP</i> , <i>svdCUIP@90</i> , <i>modSvdCUIP@100</i> using Improvement as an evaluation metric.	120

LIST OF FIGURES

4.10	Average time to generate <i>svdCUIP</i> and <i>modSvdCUIP</i>	122
4.11	Comparing the Percentage Increase of the <i>tfIdfCUIP</i> , <i>svdCUIP</i> , <i>modSvdCUIP</i> for two classes of queries: vague and self-evident.	124
4.12	Comparing the Improvement of <i>tfIdfUIP</i> , <i>tfUIP</i> , <i>tfIdfCUIP-0.09</i> , <i>svdCUIP-90-0.13</i> , <i>modSvdCUIP-100-0.63</i>	127
5.1	Partnership Ontology: concepts and properties that define relationship between them. Various other stan- dard ontologies like Dublin Core, FOAF, Geo, VCard etc are also imported.	142
5.2	Seller Profiles for this study: Seller1 and Seller2 . .	149
5.3	Seller Profiles for this study: Seller3 and Seller4 . .	150
5.4	Seller Profiles for this study: Seller5	151
5.5	An example to demonstrate construction of user pro- file (Buyer Profile) - concepts shown here are derived from the Partnership Ontology	152

LIST OF FIGURES

5.6 A reduced version of buyer profile - truncated to fit
in here. The features that buyer does not choose
during profile construction are removed to save space. 156

5.7 Search Results showing the ranked list of matching
seller profiles to a given buyer profile. 160

List of Tables

1.1	A snapshot of an exemplary <i>UIP</i> obtained from (Noll and Meinel, 2007) work on personalized search based on folksonomy	13
1.2	leftmost column shows the original rank of search results from google in middle column. Rightmost column shows the adjustment in the rank of search results after application of <i>UIP</i>	14

2.1	A comparison summary of the proposed approaches with the other similar approaches that uses folksonomy for personalized search. (a)Source of terms for constructing a <i>UIP</i> , (b) Web document Representation, (c) Similarity Measure, (d)First-Order Co-occurrence, (e) Second-Order Co-occurrence, (f)Clustering of terms in a <i>UIP</i> , (g) <i>UIP</i> and resource length normalization factor	46
3.1	Top three Hub URLs for the IMDB URL	63
3.2	Terms extracted from the <i>Hub URL</i> ₁	63
4.1	Clicked Web documents and tags attached to the documents	87
4.2	A user context derivable from Table 4.1	88
4.3	Clusters obtained by applying HAC on similarity matrices <i>Sim</i> ₃ and <i>modSim</i> ₃ for <i>k</i> =3 and <i>d</i> =0.35	100
4.4	Example of cluster structure	118
4.5	Comparing the MRRs of <i>tfIdfUIP</i> , <i>tfUIP</i> , <i>tfIdfCUIP</i> , <i>svdCUIP</i> , and <i>modSvdCUIP</i>	126

LIST OF TABLES

5.1	List of Concepts produced by amalgamating contribution of various research work's in domain of Partnership Establishment.	144
1	List of Self-evident query and target URL pairs . . .	194
2	List of vague query and target URL pairs	195

1

Introduction

The only true wisdom is in knowing you know nothing. - Socrates

Adaptive Web Systems (AWS), belongs to the class of user-adaptive software systems (Brusilovsky, Kobsa, and Nejdl, 2007) that largely depends on the existence of *user profile*. The user profile is a representation of information about a user that is essential to an adaptive system to provide the adapted effect, i.e., to recommend meaningful and relevant products or results for different users. For instance, for a user query *apple*, a search engine may return search results related to *apple* as a fruit, *apple* as an iPhone or iPad, or *apple* as in the context of eye. According to wikipedia ¹, a user profile is a collection of personal data associated with a specific user. A user profile can be manifested in different forms in different domains. What data is included in a user profile depends on the domain or the application. It may include user's interests, user's preferences, user's goals or plans, and user's likes or dislikes. To create and maintain

¹http://en.wikipedia.org/wiki/User_profile

an up-to-date user profile, a Web system collects data from various resources that may include implicitly observing user interaction or explicitly requesting direct input from the user. This process is called as **user profiling**.

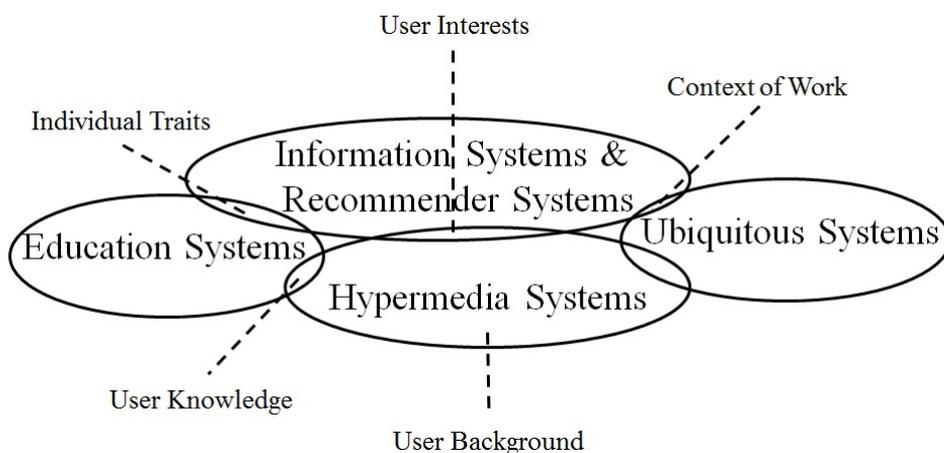


Figure 1.1: User Profiling features for various classes of Web Systems

One common feature across all Web systems is the enactment of user profiles to acculturate the system’s behaviour to individual users. User Profiles represent information about users that is essential to remodel and improve the functionality of the system with the ultimate goal of improving user experience. Web Systems have surveyed a plethora of approaches to user profiling from exploring how to accumulate user data, storing it, organizing it, and keep it up-to-date. Most of the Web systems focus on features to model information

about the users for representing a user profile. The widely used features are user knowledge, user interests, goals, background, individual traits, and context of work. Each individual Web system capitalizes on a subset of these features to model a user profile, the selection of features largely depends on the domain of interest, refer Figure 1.1. Feature based modelling of user profile aims to model user's specific features such as knowledge, interests, goals, etc. It is based on user's interaction with the system. During the user interaction, these features may change, so is the user profile. Therefore, in feature based modelling, a user profile is always up-to-date. A contrarian approach, which is an age old approach, is stereotype user profiling [163,164]. Stereotype user profiling aims to cluster all possible user types into several groups, called stereotypes. The goal of stereotype user profiling is to map individual user features to a particular group. Both methods, personalized search and partnership match, proposed in this work are based on feature based user profiling.

Since this work is focused on feature based user profiling, we will now focus on various features that are essential to building a user profile. The most widely used features are:

1. **Knowledge** : It is the most commonly used feature in Web based education systems for modelling a user profile. The user's knowledge is a variable feature, in the sense that a user's knowledge is upgrading, or deteriorating, or is staying constant. This warrants that a particular Web based education system has to recognize the changes in user's knowledge level and update the user profile accordingly. Some examples of

web based education systems that uses user profile are WITS (Okazaki, Watanabe, and Kondo, 1997), ILESA(López, Millán, Pérez-de-la Cruz, and Triguero, 1998), Web-PVT(Tsiriga and Virvou, 2003). The simplest form of user knowledge based profile is a scalar profile. It estimates the level of user domain knowledge on a scale of 0 to 5 (quantitative) or as one of the classes of good, average, fair, poor, none (qualitative). Different versions of the Web page are presented to individual users based on their levels of knowledge(Beaumont, 1994, Boyle and Encarnacion, 1998, Brailsford, Stewart, Zakaria, and Moore, 2002). The shortcoming of scalar based user profile is low precision. The user knowledge of any domain can vary for different part of the domain. It would require dividing a domain into sub-domains, and further eliciting from the user his/her knowledge of each sub-domain. These scores are then syndicated to generate a combined score for the whole domain. One of the challenge in this kind of methodology is to estimate all sub-domains for a given domain. Further, it would also require various representations of each Web page for different knowledge levels in different sub-domains. This could be a challenging task.

2. **Interests** : User Interests has had always been the most important constituent of a user profile in information systems or recommender systems that dealt with overwhelming amount of information. The personalized search methods proposed in this thesis are also based on user interests.

As a first step, the aim is to automatically identify user interests. Further, the proposed methods identify and group similar user interests into group. The similarity is identified in terms of syntactic or semantic or contextual. Early Web based education systems paid no attention to user interests. However, in the recent decade, the situation has changed dramatically. There is a competition between user interests and user knowledge when it comes to what constitutes an essential part of a user profile. This is essentially due to the increased user interactions with Web systems that are mostly interest driven, such as news systems(Abel, Gao, Houben, and Tao, 2011), electronic stores(Rossi, Schwabe, and Guimarães, 2001), museum(Rennick-Egglestone, Whitbrook, Leygue, Greensmith, Walker, Benford, Schnädelbach, Reeves, Marshall, Kirk, et al., 2011). The predominant approach to model user interests in a profile is through the weighted vector of terms or keywords, and this approach is still widely used. In contrast to keyword level approach to building user profile, another most recent approach is concept based approach to profiling user interests. Concept based approach to user profiling provides a more accurate representation compared to keyword based approach. For instance, a news personalization system can profile user interests on distinct topics, that could be based on location, genres, named entities, and so on(Abel, Gao, Houben, and Tao, 2011). In closed Web systems such as museum, even ontologies can be employed for mapping user interests to concepts in the ontologies. Whereas, in Open systems such as news personaliza-

tion, employing ontologies can be an overwhelming task. In nut-shell, open Web systems uses keyword based approach to profiling user interests and closed Web systems uses concept based approach to profiling user interests.

3. **Goals and Tasks** : User's goals and tasks represent immediate information need of a user. The user goal is most changeable user feature; it changes with each session and often changes within one session also. Planning and sequencing systems model user goals to build a user profile(Brusilovsky, 1992, McArthur, Stasz, Hotta, Peter, and Burdorf, 1988, McCalla, Bunt, and Harms, 1986, Vassileva, 1990). User's immediate information need is also diagnosed by information retrieval systems(Brajnik, Guida, and Tasso, 1987). A hierarchy of user goal is developed, and it is assumed that at one point of time user has a particular goal. This warrants identifying user goal to one of the goals in the goal hierarchy. Based on the current goal, relevant Web pages are recommended to the user or are adapted to user information needs. A popular example of goal based Web system is ADAPTS(Brusilovsky and Cooper, 2002). This system has a small hierarchy of goals. The system observes user behaviour to detect the current user goal, and depending on that, Web pages are adapted before presenting them to the user. This system was developed for aircraft maintenance operations.
4. **Background** : The user's background comprises of user's location, lan-

guage, profession, etc. For instance, clinical decision support systems can classify a user's knowledge of medical terminology to pre-defined set of categories. For each category different Decision Aids (DA) are developed. Based on the category of the user, the relevant DA is presented to the user. Another example of user adapted Web systems is the categorization of users by their language ability (native or non-native), followed by choosing the appropriate version of the content for them (Kay and Kummerfeld, 1994). Background information of a user is also used in Web based navigation support systems (Vassileva, 1996).

5. **Individual Traits** : The user's individual traits is an amalgamation of various user features that define a user as an individual. Some of the user features are personality traits (introvert/extrovert), cognitive styles (holist/serialist), cognitive factors (working memory capacity, focus), and learning styles. Similar to user background, user individual traits consist of stable features that don't change suddenly. To identify user individual traits, a psychological interview or tests are required. It has been widely acknowledged by research in IR to model user individual traits and use for personalization. Psychological literature has immense discussion with great width and depth of individual traits, however, in the field of user profiling, the interest is largely in cognitive styles and learning styles. Cognitive styles in layman terms mean an individual habit about how he/she organizes and represent information (Riding

and Rayner, 1998). Learning styles refer to how an individual learns or absorbs information. This feature is used for education based personalization systems. In the context of personalized museum guides(Krüger, Baus, Heckmann, Kruppa, and Wasinger, 2007), a user profile is used that consists of user's personality factors. Another research on adaptive Web page generation(Tarpin-Bernard and Habieb-Mammar, 2005) is based on user's lower level cognitive abilities.

6. **Context of work** : The context of work, is rather a new feature, that is being used to build a user profile in Web systems. In the beginning, it was introduced to build Web systems and later expanded into the area of personalized Web systems. In personalized clinical decision support systems, it adds a new dimension of human personal context, i.e., blood pressure, mood, cognitive load, etc. Another dimension to context in user profiling is the user platform or device, this is called as device oriented context. This kind of context is very dominant in mobile and ubiquitous computing. Finally, one more dimension that has been added to context is the context of work. This is the context of work that the user is dealing with. It is called a user oriented context. For instance, in the partnership match, we have taken the context of work as user profiling. Since the context of work is to find partners, and partners are represented as buyers and sellers, therefore, we have modelled two different types of profiles: buyer profile and seller profile.

1.1 User Profiling for Personalized Search

In this thesis, refer Figure 1.2, a user profile is manifested as User Interest Profile and buyer profile or seller profile in the domain of Personalized Search and Partnership Match, respectively. Chapter 3 and 4 demonstrates how a user profile is manifested as a User Interest Profile (*UIP*) for personalized search. Chapter 5 demonstrates how a user profile is manifested as a buyer profile or seller profile for partnership match



Figure 1.2: User Profiling for Personalized Search and Partnership Match

1.1 User Profiling for Personalized Search

A search engine returns the most relevant search results matching a user query, however, it often fails to judge the user query intent or user interests. To improve the quality of search results, the system needs to understand different aspects associated with a user query: one is user interest, and the other is query intent. A user model, built from user interactions with the Web and folksonomy, plays a bigger part in disambiguating query intent by taking clues from user interests. User interests can be considered as contextual variants that may help to disambiguate user query intent when the original query is vague or there are too many search results that a user has to wade through to find the most relevant ones. Moreover, the amount of information available on-line is

1.1 User Profiling for Personalized Search

increasing exponentially. While this information is a valuable resource, its sheer volume limits its value. Many research projects and companies are exploring the use of personalized applications that manage this deluge by tailoring the information presented to individual users. These applications need to gather, and exploit, some information about individuals in order to be effective.

1.1.1 Motivation

The most prevalent way for computer users to find the required information is to surf the Web and search through Internet pages. Having various available free search engines, such as Google, Bing, and Yahoo, makes Internet searching the first and easiest way to find relevant content. In this case, the user expresses his information need as a small set of keywords and receives a ranked list of documents. Having a list of retrieved documents, however, is not enough for the user to find the exact information that he is looking for. The user has to spend more time with these documents to extract the exact information need from the large retrieved documents. Such a manual processing step is not possible without spending a large amount of time.

Personalization has emerged as an appealing approach when dealing with the issues caused by the variation of on-line behaviors and individual differences observed in user interests, information needs, search goals, query contexts, and others (Ioannis, Konstantinos, and Joemon, 2010). Personalized Search Engines return different results for different users even though the input query is same. The results are differentiated based on the input query by the user

1.1 User Profiling for Personalized Search

and user interests. In certain scenarios, search results are re-ranked based on each user interests. These leads to improved search quality, and it needs additional efforts which indicates that developing a personalized search system needs studies beyond search engine development. This goal is mainly achieved using a combination of important techniques:

1. **Natural Language Processing** methods which analyze input documents and user search history to build user profile.
2. **Information Retrieval** methods which retrieve a set of relevant documents from the input corpus and re-rank them based on the user profile
3. **Data Mining** methods which clusters the terms in the user profile so that contextually similar concepts are grouped together thus disambiguating polysemy and synonymy. Also, it requires matrix factorization methods to discover latent information that is useful to calculate the similarity between terms in the user profile.

To achieve such a system, a pipeline of different components is required which constructs the whole architecture of the personalized search system. This dissertation mainly focusses on building such system.

1.1.2 Research Problems

To achieve a personalized system, one of the core requirement is to build a profile of user interests. Existing research works in user modeling use the phrase

1.1 User Profiling for Personalized Search

user profile which can be misleading; a *user profile*¹ often means user personal information, such as name, address, and age. Our intention is not to collect user personal information, instead, our goal is to collect user interests. We, therefore, coin a new term, User Interest Profile (*UIP*), which we believe is more appropriate because such a profile reflects user interests and not user personal information.

The primary research problem, addressed in Chapters 3 and 4, is building a User Interest Profile (*UIP*) that consists of user interests and their context. The *UIP* is further used for re-ranking search results, thus providing personalized results to a user. User interests are inferred from user search behavior which is obtained by mining user's search history or URLs clicked by the user during his/her search sessions. Given a list of clicked URLs, interesting re-search problems are: How to summarize them to generate a list of terms, How to eliminate noisy terms, and How to determine context of terms that represent user interests?

Most recent works, (David, Iván, and Joemon, 2010, Noll and Meinel, 2007, Xu, Bao, Fei, Su, and Yu, 2008), related to personalized search use folksonomy to build a *UIP* from the clicked web pages; however, there are some inherent limitations which we discuss next, and propose solutions to remedy them.

Limitation 1: The concepts, that make a *UIP*, are collected from the re-source profiles of clicked URLs emanating from user search sessions. A *UIP*

¹http://en.wikipedia.org/wiki/User_profile

1.1 User Profiling for Personalized Search

is further used in other search sessions to re-rank search results by calculating cosine similarity between the resource profiles of search results URLs with all concepts in a *UIP*. To ease the exposition, consider a scenario from (Noll and Meinel, 2007) work on user profiling for personalized search. Table 1.1 shows the *UIP*, for a user, constructed using folksonomies.

Table 1.1: A snapshot of an exemplary *UIP* obtained from (Noll and Meinel, 2007) work on personalized search based on folksonomy

Concept Name	Concept-Weight
open source	13
programming	19
proprietary	2
research	10
security	21
semantic web	34

1.1 User Profiling for Personalized Search

Table 1.2: leftmost column shows the original rank of search results from google in middle column. Rightmost column shows the adjustment in the rank of search results after application of *UIP*.

Original Rank	URL	Re-ranked
1	securityfocus.com/	1 •
2	microsoft.com/security/	7 ↓
3	microsoft.com/technet/security/def/...	3 •
4	dhs.gov/	10 ↓
5	whitehouse.gov/homeland/	9 ↓
6	windowsitpro.com/WindowsSecurity/	8 ↓
7	ssa.gov/	5 ↑
8	w3.org/Security/	4 ↑
9	cert.org/	2 ↑
10	nsa.gov/	6 ↑

One can infer from the *UIP* in Table 1.1 that the user interests are *security*, *programming*, *research*, and *semantic web*. Table 1.2 shows the effect of *UIP* on the ranking of search results. The leftmost column of Table 1.2 shows the original ranking of search results returned by the Google search engine for a user query *security*. The rightmost column of Table 1.2 shows the adjustment in the ranks of the search results after personalization based on *UIP* in Table 1.1. Meticulously observing the leftmost column and the rightmost column

1.1 User Profiling for Personalized Search

of Table 1.2, one can infer that the URLs related to terms *computing* and *security* are promoted to the top. However, there exists no reasoning that explains the quantitative effect of the terms, in a *UIP*, on the ranking of search results. That is, why a particular URL gets promoted more than the other URL, when both the URLs are relevant to the same term, say term *security*. Or, why a URL is promoted more than the other URL, even though one of the URL is less related to the user query compared to the other URL. Authors mention that the URL of US Security and Administration is promoted even though it is not related to concepts *computing* and *security*. We offer the following explanation; some terms, in a *UIP*, even though not related to user query *security*, but because they are present in a *UIP*, contributes to the ranking score of URLs in the search results. The term, in this case *insurance* in the *UIP* (not shown in Table 1.1 but authors mentioned in their paper that concept *insurance* exists in the *UIP*), has a false positive effect on the ranking of URL. The reason, why the URL of US Security Administration is promoted, is because of the incapability of the system to judge the context of user query *security*. Note that, the terms, in a *UIP*, may have false positive or false negative effect on the re-ranking of URLs, which is actually uncalled for. We claim that the related terms in a *UIP* should be clustered together and work as a cluster; since *security* and *insurance* are unrelated terms, URLs that are re-ranked based on the term *security* should not be effected by the presence of term *insurance* in a *UIP*. In other words, the term *insurance* should not contribute towards the re-ranking score of the search results obtained from a

1.1 User Profiling for Personalized Search

search engine for a user query *security*. The terms, in a *UIP*, that are related to concept *security* can definitely help to disambiguate it, for ex: if a item *IT* is clustered together with a term *security*, and both are used in conjunction for computation of re-ranking score with the resource profiles of search results; the computed re-ranking score will help to positively promote the rank of URLs related to terms *IT* and *security*, and demote the rank of URLs related to terms *security* and *device*, or *security* and *administration*, or alike. In the existing work, terms in a *UIP* are not clustered into groups, therefore whether the terms are related to a user query or not, they anyway participate in the computation of re-ranking score. We propose to cluster the related terms in a *UIP* resulting in a clustered *UIP*. A cluster of terms, related to a user query instead of all terms in the *UIP*, is used for calculating the re-ranking score of URLs. This allows to consider terms in a matching cluster to a user query for re-ranking score computation with the resource profiles of search results.

The experiment results verify our claim that clustering the terms, present in a *UIP* thus generating a clustered *UIP* (*CUIP*), has many advantages; it helps to disambiguate context of a user query, mitigate polysemy problem and synonymy problem, reduces the time complexity of re-ranking, and improves the precision of the search results. The clustering of concepts in a *UIP* allows to disambiguate user interests by associating the context which is otherwise latent.

Limitation 2: A resource like URL is tagged by many users. For each URL, a resource profile is created. But since, users don't tag resources religiously;

1.1 User Profiling for Personalized Search

it may be possible that a resource profile, of a particular URL, has tags with higher tag-weights while others don't. Popular URLs, compare to less popular URLs, are tagged by many users. Hence, popular URLs have more number of tags with high value of tag weights. The existing work does not take into account the biases of tagging by users. To alleviate such biases, we propose to normalize the value of tag weights associated with tags in a resource profile. For illustration purpose, consider resource profiles of two URLs: $URL_1 = \{java : 50, programming : 10\}$, $URL_2 = \{java : 5, programming : 1\}$. The resource profile of URL_1 and URL_2 have similar tags, but tags in resource profile of URL_1 have higher value of tag-weight. Existing work is based on the hypothesis that value of tag-weight reflect the importance of tags in a *UIP*. However, if we normalize the tag-weights of tags in a *UIP*, it gives a different picture. After normalization, resource profile of URLs will be as follows: $URL_1 = \{java : 5, programming : 1\}$, $URL_2 = \{java : 5, programming : 1\}$. This suggests that both tags are equally important for URL_1 and URL_2 .

Limitation 3: We experimented with the search query log of users and observed that users exhibit sporadic search behavior. We find that two factors, viz. user search behavior and URL popularity, effect the number of tags and value of tag-weights in a *UIP*. This further means that, some users search actively while others are intermittently active. Active users' *UIP* consists of tags with high value of tag-weights while non-active users' *UIP* contains tags with low value of tag-weights. Existing works, assume that, users whose *UIPs* have

1.2 User Profiling for Partnership Match

tags with high value of tag-weights are more interested in those tags. While non active users whose *UIPs* have tags with low value of tag-weights are less interested in those tags. The biases of user search activity can lead to invalid personalized search results(Wang and Jin, 2010). We propose to annul or dilute the biases due to sporadic user search behavior by normalizing the tag-weights in each *UIP*.

1.2 User Profiling for Partnership Match

In order to maximize the advantages and minimize the negative effects of globalization and growing interdependence, it is imperative for SMEs (Small and Medium Enterprises) in developing countries to forge partnerships with big enterprises in developed regions. However, the partnership establishment process is a rough ride; it comes with its own set of hurdles. A survey by PricewaterhouseCoopers (PwC) indicates that 44% of the partnerships were unsuccessful. In this dissertation, we refer to research literature to find out various features that are involved during partnership establishment process. Based upon a review, we select features that form core concepts in a partnership establishment process. These concepts along with their related properties are modeled as an ontology, termed as Partnership Ontology. A user that could represent a big enterprise or a SME (Small and Medium Enterprise) can use the partnership ontology to lay down their requirements as a buyer profile and/or a seller profile. A semantic similarity measure is defined to compute a ranked list of matching

1.2 User Profiling for Partnership Match

seller profiles given a buyer profile. We illustrate the devised methodology of partnership establishment process by an example using a case study.

1.2.1 Motivation

Partnership is a voluntary collaborative agreement between two or more parties in which all participants agree to work together to achieve a common purpose or undertake a specific task and to share risks, responsibilities, resources, competencies and benefits. Meaningful partnerships are the foundation for success. Partnerships are what enable many companies to make continuous improvements. By sharing with others, one can direct their resources and capabilities to projects what they consider most important. The selection of the right partners is a critical element of an Extended Enterprise (EE) strategy. Although most companies understand the importance of selecting the right partner, they often do not spend enough time understanding their individual needs and defining their requirements. As a result there is a greater risk of an incorrect selection decision, which may ultimately lead to a failed partnership. This has negative ripple effect for other parties along the EE from down through the supply chain and forward through the customer chain. A survey taken by Business Consultants has revealed that 49% of the partnerships are very successful, 44% results in partial success and 7% are a failure, shown in Figure 1.3. The most common causes of failure cited by CEOs are: cultural differences, poor or unclear leadership, and poor integration process. The above are the major reasons, though there is plethora of factors that affect a partnership establishment process.

1.2 User Profiling for Partnership Match

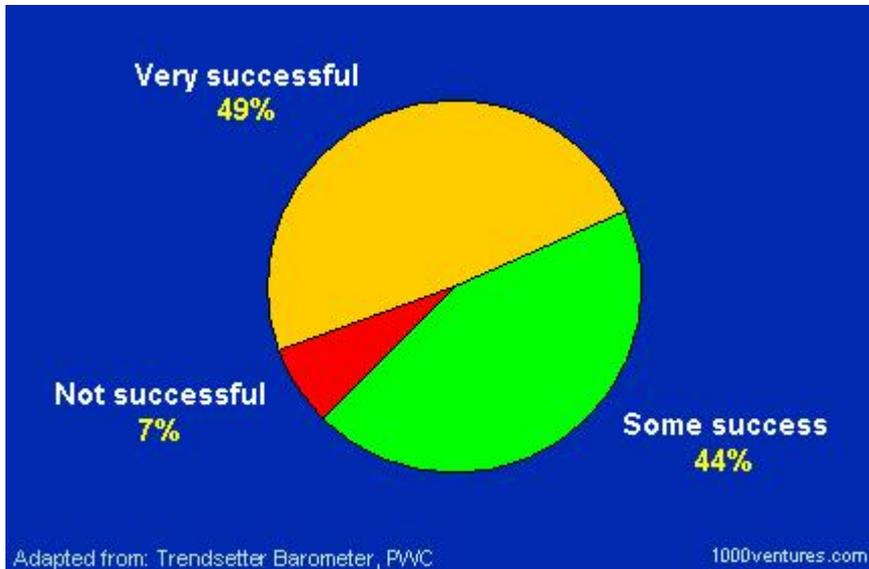


Figure 1.3: Percentage of partnerships that are successful, partial successful, and failures

Figure 1.4 below shows that 49% of the failures are due to poor or unclear leadership, another 49% are due to cultural differences, whereas 46% of the failures are due to poor integration processes. Analysis of these results gives enough reason to improve the partnership matching process so as to reduce the partial success and failure partnerships. Another survey carried out by PwC (PriceWaterHouseCoopers) interviewed CEOs of 239 Fortune 500 companies, refer Figure 1.5; results from the survey shows that 56% of the companies in US have partnered over the past 3 years. These companies have partnered with large companies (41%), large MNCs (28%), large domestic companies (22%),



Figure 1.4: Reasons that cause failure of partnership

small companies (29%), university (7%), and federal lab (3%). The interviewees cite three major benefits of partnering, based upon their own experiences: increased profit opportunities (88%), secured competitive position (87%), and increased sale of existing products (80%), refer Figure 1.6. Two other benefits are creation of more new products or lines of business, cited by 66%; and better operations or technologies (60%). The emergence of globalization process in industrial scenario is forcing users to consider forming network partnerships and collaborations, such as EE, in order to achieve a sustainable competitive advance and growth. However, the success rate of partnerships is found to be low, which is due to the selection of unsuitable partners. Therefore, part-

1.2 User Profiling for Partnership Match

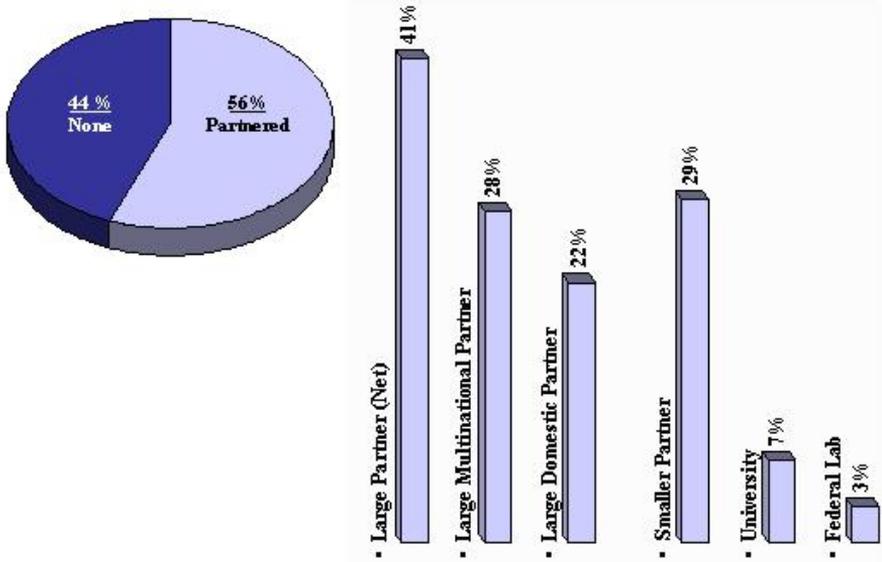
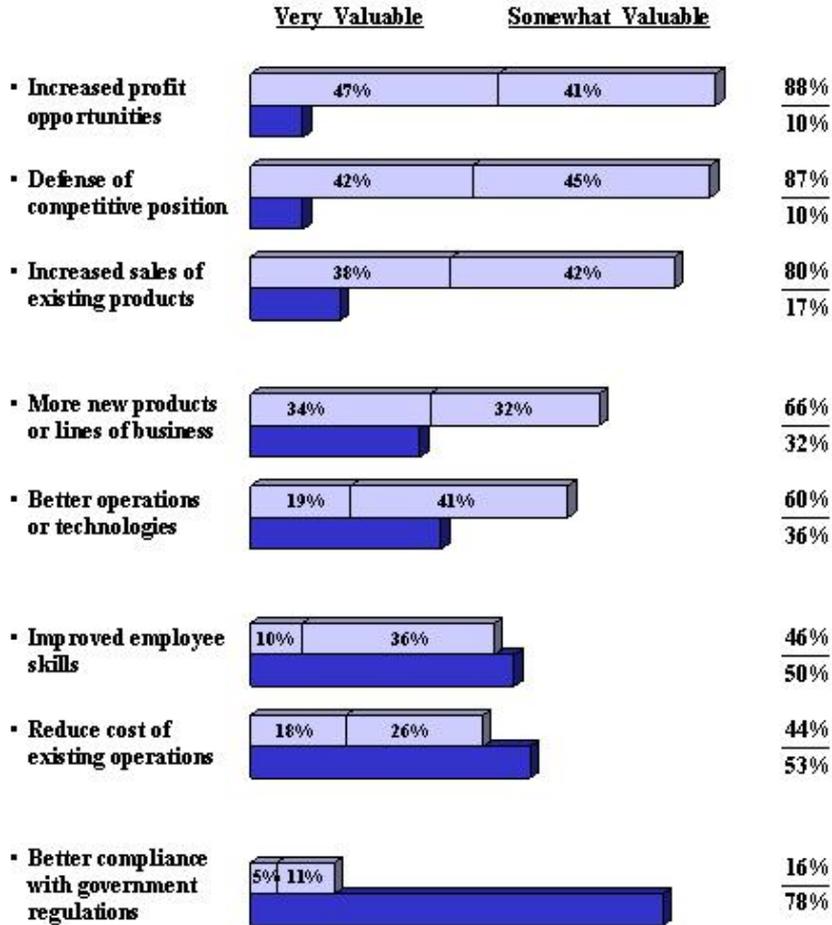


Figure 1.5: Percentage of companies who has formed joint ventures with other companies.

nership match plays a key role in the success of a partnership. A meticulous examination of the key components in the partnership match reveals that a very few formal partnership match process exist, and those that do are not sufficient to support partnership match effectively; results in Figure 1.3 vouch the said claim. This is further complicated when an ODM from a developed country, for instance South Korea, seeks a partner from a developing country, such as India. Thus a critical question is how globally separated organizations can be supported to establish an EE partnership that increases the chances of the optimum set of partners being selected, while being conducted effectively

1.2 User Profiling for Partnership Match

and efficiently.



Base: 239 Firms Involved in Partnering Over Past 3 Years

Very or Somewhat Valuable
 Not Too or Not At All Valuable

Figure 1.6: Key benefits of partnering.

1.2.2 Research Problems

The projects that operate within inter-enterprise environments additionally face the problem that different information models are likely to be used by different partners. Engineers working within a particular organization will inevitably develop their own vocabulary for particular activities and these will need to be adjusted to be more practical and to meet the requirements of different collaborating partners. Hence, when two different partners are brought together, two common types of problem can occur in communications that share and exchange information, firstly, the same term is being applied to different concepts (semantic problem), secondly, different terms may be used to denote the same entity (syntax problem). This problem is popularly known as integration problem in literature.

The objective of the proposed Partnership Match is to explore the fundamental problem: How distributed organizations be supported to establish an EE (Extended Enterprise) partnership that increases the chances of the optimal partner being selected, while being conducted efficiently and effectively without any syntax or semantic disambiguation.

The key hypothesis of partnership match is that, a process perspective is employed in order to help users representing organizations effectively manage their distributed partnership establishment process. This structured approach enables both users and associated users' profile information to be presented in a generic machine readable format, a mechanized matching process to take place

and partnership management to be managed effectively.

In order to explore such hypothesis, this thesis intends to answer several problems: how to effectively model user's profile; i.e. what should be the key components that form a user profile, how to make user profile machine readable so that it can be processed and further reasoned by the machine, and to define semantic similarity measures for compare user profiles.

By solving these problems, Partnership Match will allow the development of new services to manage social interactions, establishing a partnership process between users (buyers and suppliers), creating a conducive collaboration environment, and a structured approach to managing the generation, and machine to machine manipulation, of request and offer profiles as part of partnership match process. These services will open new business opportunities for networked enterprises to provide new products/services. Partnership Match will develop generic services, applicable across different domains, and specifically explore new business opportunities in manufacturing and engineering SMEs.

1.3 Contributions

The main contribution of this dissertation is to improve user satisfaction in the context of search results and partnership match.

To this aim, for personalized search, we propose three methods to model user profile and also propose an automatic evaluation method. And, for partnership match, we propose an ontology that can be used for building user profiles that

can be further modelled as buyer profiles or seller profiles.

1. The first method for personalized search is a non-folksonomy based method. It is called as Exclusively Yours'. It uses anchor text to build a *UIP*. We also propose how to compute term-weights for terms in the *UIP* and also how to find matching terms in a *UIP* for a given user query.
2. The second method for personalized search is a folksonomy based method. It uses Singular Value Decomposition (SVD), a matrix factorization method to discover latent information, to generate a *svdCUIP*.
3. The third method for personalized search is also folksonomy based method. It is a variation of SVD, modSVD, to generate a *modSvdCUIP*. *modSvdCUIP* represents a better cluster structure as compared to *svdCUIP*.
4. One of the impediments in the personalized search research area is evaluation. Researchers find it difficult to get access to user query logs, and even if they can get access to it, evaluation also requires users' involvement to evaluate the quality of search results. We propose an automatic evaluation method that doesn't requires user involvement at any stage. Thus our proposed methods, or for that matter, any personalized search method can be evaluated using our proposed evaluation method.
5. For partnership match, I propose an ontology to provide a machine readable representation of buyer and seller profiles. A semantic similarity

measure is also proposed that ranks seller profiles for a given buyer profile. The system is implemented as a web service that can be hosted on a web server, thus providing an easy access to users. The proposed methodology is unique in the sense that ontologies are employed and vector space model is used so as to provide a solid systematic approach which is also mathematically proven. The major innovation of the proposed methodology is that the UNSPSC ontology provides a unique code for manufacturing skills that helps in disambiguation of any product or services. Classifying products and services with a common coding scheme facilitates commerce between buyers and sellers and is becoming mandatory in the new era of electronic commerce.

The existing works, for construction of a *UIP*, assume that a user is registered with one or more social network service. We don't make such assumption. The proposed system observes and analyzes a user search behavior to construct his/her profile. Thus our system is applicable to all users with no dependency on a particular search engine or a particular social network service (SNS). The system architecture developed in this work can be used with any search engine or any SNS, provided the search engine or SNS has its open access API available.

In addition to the proposed methods for building a *CUIP*, we also propose an automatic evaluation method to test the proposed methods with the baseline search and folksonomy based personalized search approaches. In our evaluations, we found that the improvement in the ranking scores of the target URLs

1.3 Contributions

for the *modSvdCUIP* based personalized search were better than all the other methods; the *modSvdCUIP* approach showed improvement of 71.6%, 27.8%, 12%, 6.6%, and 8.1% over the baseline (Lucene Search), *tfIdfUIP*, *tfUIP*, *tfIdfCUIP*, and *svdCUIP* approaches, respectively.

1.4 System Architecture - Personalized Search

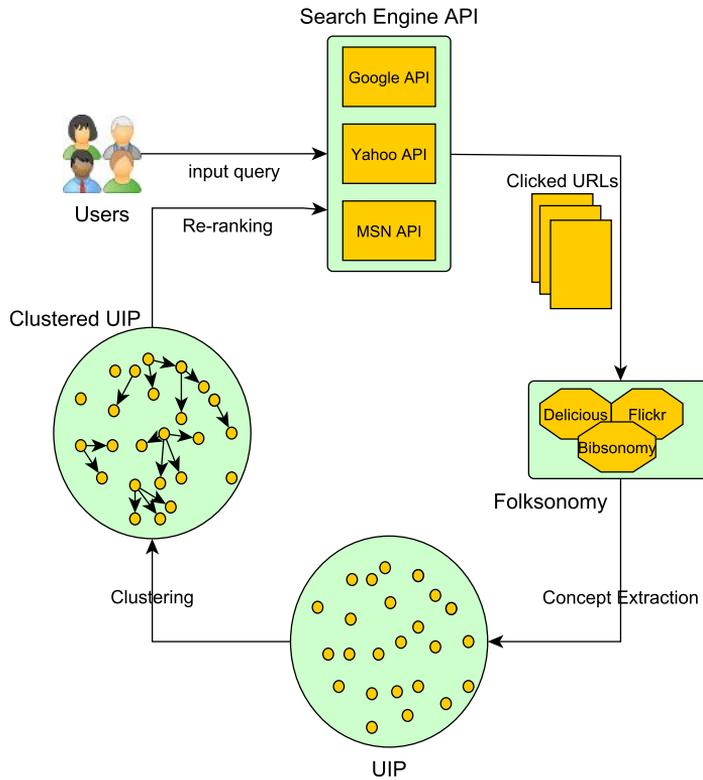


Figure 1.7: A system architecture for building a *CUIP* and its application to personalized search

This section describes the system architecture for building Clustered User Interest Profile (*CUIP*), and how the *CUIP* is used for re-ranking search results for personalization. It begins with the explanation of the sequence diagram

1.4 System Architecture - Personalized Search

that encompasses various modules of the system; collecting user search history, extracting and mining user interests from user search history to build a *UIP*, clustering concepts to build a *CUIP*, and finally using *CUIP* for personalized search. Figure 1.7 shows various modules and their connections using a sequence diagram.

A user session begins with a given input query. The input query is submitted to a search engine, and the output is a ranked list of URLs. Furthermore, based on the relevancy of the output ranked list of URLs, a user clicks on URLs of his/her interest. A list of clicked URLs, which we believe reflect user interests, is processed to extract concepts. To extract concepts for a given URL, it is submitted to a social bookmarking service which returns a list of tags and tag-weights. The list of tags and tag-weights are imported to construct a *UIP*. The extracted terms are further manipulated using factorization techniques and clustering algorithms to discover a set of meaningful concept clusters. The final clusters of terms represent a *CUIP*. Each concept in a cluster has a weight associated with it reflecting its importance in the cluster. The *CUIP* is further used for re-ranking search results to provide a personalized search result set for a given input user query in the following search sessions. Figure 1.7 shows three search engine APIs: Google API, Yahoo API, MSN API; this only means any one of the API can be used to obtain search results. Similar reasoning goes for the folksonomy.

1.5 System Architecture - Partnership Match

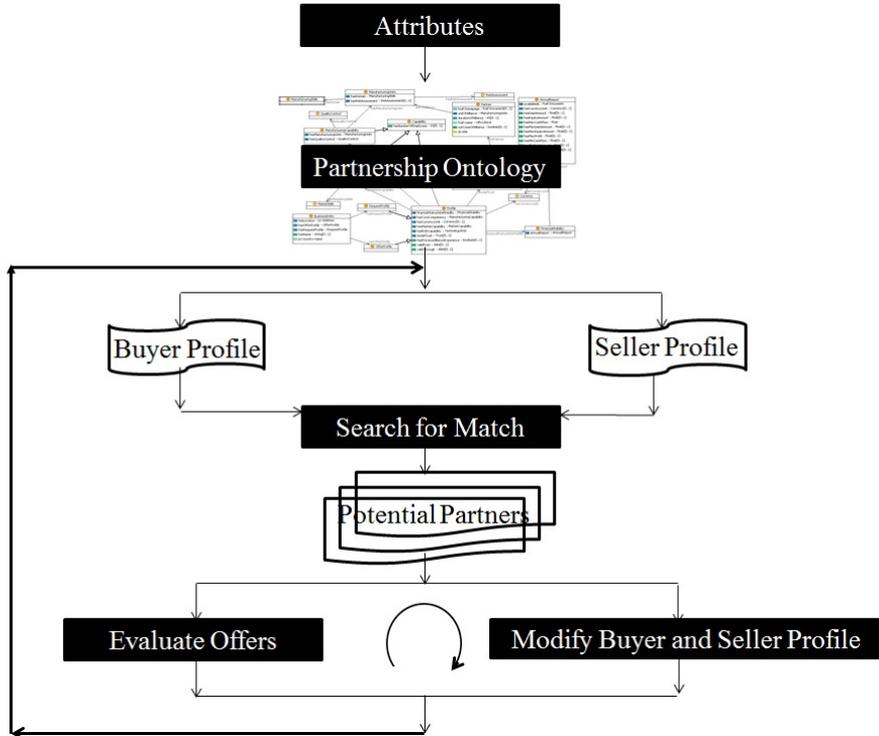


Figure 1.8: An archetype for Partnership match, showing the flow of processes

The architecture, shown in Figure 1.8, is developed in such a way that it prompts the user to adopt a systematic approach to partnership establishment. A web enabled software prototype is developed and used to validate the architecture. The success of partnerships establishment is significantly influenced by the manner in which profiles are created. A profile is simply a set of

1.6 Organization of this Dissertation

generic facts about a user requirements representing an enterprise, which may be used by other users to determine their suitability as potential partners. A seller profile records the capabilities and capacity of the potential partner. A buyer profile is a mechanism utilized to communicate to the SME what the potential partner can do to meet their needs. The first step of any partnership establishment process should take place with both the parties defining their terms (requirements and offer attributes). A user looking for SME partners makes a buyer profile; whereas, SMEs make a seller profile, note that both are oblivious of each other, i.e., they just make their profiles available to the system. Buyer, after providing his profile, searches for the matching seller profiles, which the system recommends after executing a semantic similarity match among various profiles available to the system. The result from searching is a set of possible partners that a buyer can consider to be his/her future partners. At this stage a buyer communicates his interest to the potential SME partners and negotiates by modifying his profile. In other words, profiling is acting as a communication channel between users. The next and final step is to select one of the SME partners from the list of available partners after negotiations and proceed with face to face meetings, discussing contract details, etc.

1.6 Organization of this Dissertation

To start with, so as to put the contributions in perspective, the Chapter 2 presents a through survey on relevant research topics. The topics include search

1.6 Organization of this Dissertation

engines, user profiling, matrix factorization, clustering, ranking algorithms, and related folksonomy based personalized search algorithms. The main contribution of this dissertation start with Chapter 3. In this chapter, a novel approach to construct a user profile, called as User Interest Profile (*UIP*) in this dissertation, from user interactions with the web is presented. It capitalizes on the user's search history and link structure of the web that includes anchor tags to build a *UIP* and use that for personalized search. In the next chapter, chapter 4, I explore folksonomy based approaches to construct *UIP* and *CUIP*. Two methods are presented that leverage upon the folksonomy to build a profile of user interests, called as *UIP*. The *UIP* is further processed using matrix factorization algorithms to extract hidden semantics in it so as to group related tags together that could be either syntactically related, semantically related, or contextually related. To group these related tags together into clusters, thus generating a Clustered User Interest Profile (*CUIP*), where each cluster identifies a unified topic, clustering algorithms are used. For the non-folksonomy based approach, one custom data set is used, and it compared the proposed method with other non-folksonomy based methods. Two different data-sets were constructed for the evaluation of folksonomy based methods for personalized search: twitter data-set and AOL query log . The twitter data set was established to evaluate the sparsity of information in *UIPs* and *CUIPs* and to test the clustering tendency and clustering accuracy of *CUIPs*; AOL data-set, which is a much larger data-set of user search histories, was harvested from AOL Search Query Log. This data set was used to test the improvement in

1.6 Organization of this Dissertation

personalized search for the two proposed folksonomy based methods and to compare them with other folksonomy based personalized search methods. In Chapter 5, I propose a partnership ontology that is used for building buyer profiles and seller profiles. A web service is developed that can be used by users for representing their respective profiles, and it also allows to find matching seller profiles for a given buyer profile. I conclude in Chapter 6 with summarizing remarks, a discussion on directions that the presented research topics can take here-on, and a general discussion on the future work.

2

Background

*The warrior who trusts his path doesn't need to prove the other is wrong. -
Paul Coelho*

We have witnessed great interest and a wealth of promise in content-based document retrieval as an emerging technology in the last decade. While a firm foundation has been laid, it also paved the way for a large number of new techniques and systems, got many new people involved, and triggered stronger association of weekly related fields. In this chapter, we survey key theoretical and empirical contributions in the current decade related to Social Semantic Web, Search Systems, User Profiling, Personalization, and Partnership Match.

2.1 Introduction to Social Web

The Social Web is an ecosystem of participation, where value is created by the aggregation of many individual user contributions (Tom, 2008). The Social Web is represented by a class of web sites and applications in which user

2.1 Introduction to Social Web

participation is the primary driver of value. The architecture of such systems is well described by Tim O'Reilly (Tim, 2005), who has fostered a community and media phenomenon around the banner of Web 2.0. Headliners for the festival include Wikipedia, MySpace, YouTube, Flickr, Del.icio.us, Facebook, and Technorati. Discussions of the Social Web often use the phrase "collective intelligence" or "wisdom of crowds" to refer to the value created by the collective contributions of all these people writing articles for Wikipedia, sharing tagged photos on Flickr, sharing bookmarks on Del.icio.us, or streaming their personal blogs into the open seas of the blogosphere. Tagging has become a valuable feature for organizing such resources. The potential for knowledge sharing today is unmatched in history. Never before have so many creative and knowledgeable people been connected by such an efficient, universal network. The costs of gathering and computing over their contributions have come down to the point where new companies with very modest budgets provide innovative new services to millions of on-line participants. The result today is incredible breadth of information and diversity of perspective, and a culture of mass participation that sustains a fountain of publicly available content.

Collective intelligence is a grand vision, one to which I subscribe. However, I would call the current state of the Social Web something else: collected intelligence. That is, the value of these user contributions is in their being collected together and aggregated into community- or domain-specific sites: Flickr for photos, YouTube for videos, etc. I think it premature to apply term collective intelligence to these systems because there is no emergence of truly new levels

2.1 Introduction to Social Web

of understanding. From the Social Web collective we can learn which terms are popular for tagging photos or the buzz in the latest blog posts, and we can discover the latest new talent in video, photography, or op-ed. However, while popularity is one measure of quality, it is not a measure of veracity. Mass authoring is not the same thing as mass authority. Particularly in the presence of spam and other fraudulent sources in the mix, simply collecting the contributions of the masses does not lead to new levels of intelligence.

Collective intelligence has been the goal of visionaries throughout the history of the Internet. Douglas Engelbart, who invented groupware, the mouse, and a form of hypertext designed for collective knowledge, wrote in 1963 of his career and project objective: "The grand challenge is to boost the collective IQ of organizations and of society" (Engelbart, 1962). His Bootstrap Principle was about a human-machine system for simultaneously harvesting the collected knowledge for learning and evolving our technology for collective learning. In human-machine systems, both the human and machine contribute actively to the resulting intelligence, each doing what they do best. Other early pioneers of the human-machine model of collective intelligence include Norbert Wiener, the father of cybernetics, Buckminster Fuller, the consummate inventor and system thinker, and Stewart Brand, creator of the first large virtual community on the Internet (Fred, 2006).

The key, as the visionaries have seen, is a synergy between human and machines. Clearly, there are different roles for people and machines. People are the producers and customers: they are the source of knowledge, and they have

real world problems and interests. Machines are the enablers: they store and remember data, search and combine data, and draw mathematical and logical inferences. People learn by communicating with each other, and often create new knowledge in the context of conversation. The Internet makes it possible for machines to help people create more knowledge and learn from each other more effectively. With the rise of the Social Web, we now have millions of humans offering their knowledge on-line, which means that the information is stored, searchable, and easily shared. The challenge for the next generation of the Social and Semantic Webs is to find the right match between what is put on-line and methods for doing useful reasoning with the data. True collective intelligence can emerge if the data collected from all those people is aggregated and recombined to create new knowledge and new ways of learning that individual humans cannot do by themselves.

The Social Web reflects that more and more Web systems accomplish an architecture of participation, which involves participation of end-users. Resource sharing systems like Flickr or YouTube depend on their users, who contribute pictures and videos, because the main purpose of these systems relies in sharing user-contributed content. Social tagging supports resource sharing within these systems (Hotho, Jäschke, Schmitz, and Stumme, 2006): "social resource sharing systems are Web-based systems that allow users to upload their resources, and to label them with arbitrary words, so-called tags". For example, in Flickr a user may publishes pictures from her latest travel to France, which she annotates with keywords such as *france*, *Paris* or *beautiful-nature*. These

2.1 Introduction to Social Web

tags will help the user to retrieve certain images in the future and therewith support her and others, as we capitalize in this work, personal information management (Heckner, Heilemann, and Wolff, 2009). Further, other users will be enabled to find the pictures if they utilize the corresponding tags to search for Flickr pictures (Kumar and Kim, 2012, 2011, Lee, Kim, Shin, and Kim, 2009, Sigurbjörnsson and Van, 2008).

Social tagging does not require pre-defined taxonomies, but vocabularies used for organizing resources in tagging systems rather emerge like desire lines (Schmitz, 2006). The structures that emerge from social tagging are called folksonomies. The term folksonomy was first introduced by Thomas Vander Wal (Vander, 2005, Feb 2007) and depicts the structures that evolve over time when users (the folks) annotate resources with freely chosen keywords. Folksonomies relate users, tags and resources based on the tag assignments that are performed by the user community. Tag assignments are triples that state which user assigned which tag to which resource. Hence, a folksonomy can thus be considered as a collection of tag assignments and folksonomy systems are those systems that allow for the evolution of folksonomies.

Today, there exist many diverse folksonomy systems in various domains. For example, Last.fm enables users to annotate music, bookmarks can be tagged in systems such as Delicious, BibSonomy supports social tagging of research articles, Amazon enables their customers to tag products, and Google Mail users can organize their emails via freely chosen labels.

2.2 Matrix Decomposition Methods

Data Mining is about finding new and interesting information from data (Jiawei and Micheline, 2001). The underlying assumption is that there is too much data for a human to process, and thus one needs an automated method that can process the corpus and find interesting and relevant information. Given the huge amount of data, it is computationally time consuming job to execute data mining or machine learning algorithms on them. Matrix decomposition methods are executed as a pre-processing step where the objective is to filter out less relevant information and only keep the more relevant ones.

Matrix decomposition, where a given matrix is represented as a product of two or more matrices, are regularly used in data mining. Most matrix decompositions have their roots in linear algebra, but the needs of data mining are not always those of linear algebra. One of the basic concept of Matrix decomposition algorithms is a matrix. In linear algebra, an n -by- m matrix is usually interpreted as a linear map from n -dimensional space to m -dimensional space (Gene and Charles, 1996). But, in data mining, and also in this dissertation, matrices are a convenient way to store and manipulate data. We have used matrices for storing text documents as term frequency matrices (Jiawei and Micheline, 2001).

Every matrix decomposition has three concepts related to it. First of these is the formulation of decomposition, that is, to what kind of matrices the decomposition applies (example, only to non-negative matrices or only to binary

2.2 Matrix Decomposition Methods

matrices), and what kind of factor matrices are feasible for the decomposition (example, non-negative matrices or orthogonal matrices). Second concept is the concrete decomposition of some matrix \mathbf{A} . Third concept is the problem of finding a decomposition that admits the formulation, given some matrix \mathbf{A} . When performing a matrix decomposition on some matrix, it is represented as a product of two or more factor matrices. The most widely used method to decompose a matrix is the Singular Value Decomposition (SVD)(Gene and Charles, 1996). It decomposes a matrix A into the form $U \Sigma V^T$, where U and V are orthogonal matrices, that is $U^T U = V^T V = I$, and Σ is a diagonal matrix with non-negative entries - the singular values of A . The Singular Value Decomposition gives the optimal rank- k approximation of the original matrix A . The optimal rank- k approximation of A can be obtained from its Singular Value Decomposition by setting all of the k largest singular values to 0. Computing the SVD is also relatively fast; it can be done in time $O(\min n^2 m, n^2 m)$ for n -by- m matrices (Gene and Charles, 1996). The methods often employed in practice, such as Lanczos methods (Gene and Charles, 1996), are usually even faster. Nevertheless, for extremely large matrices that can still be overwhelming. This has motivated the study of fast, approximate decomposition algorithms that are based on sampling the original matrix. Work done in this field include the results of (Alan, Ravi, and Santosh, 2004), (Drineas, Kannan, and Mahoney, 2006a), (Drineas, Kannan, and Mahoney, 2006b), (Drineas, Mahoney, and Muthukrishnan, 2006c), (Drineas, Mahoney, and Muthukrishnan, 2006d), (Drineas, Mahoney, and Muthukrishnan, 2008), and (Achlioptas and

2.2 Matrix Decomposition Methods

McSherry, 2001).

If a matrix A is non-negative (example, because it is a result of measurements that can only yield non-negative results), interpreting the results of SVD can be problematic. This is because for a non-negative matrix A , the U and V factor matrices produced by SVD can contain non-negative values. This problem is addressed by Non-negative Matrix Factorization (NMF) methods, where the factor matrices are required to have only non-negative values. Early formulations of the NMF problem include (Paatero and Tapper, 1994), where they called it ‘positive matrix factorization’, and (Cohen and Rothblum, 1993). However, the most famous is due to (Lee and Seung, 1999). Since their article, the problem has attracted a lot of attention and many researchers developed an innumerable number of algorithms (Berry, Browne, Langville, Pauca, and Plemmons, 2007).

In addition to SVD and NMF, many other matrix decomposition algorithms have been proposed, most of which are based on probabilistic models. Such methods include multinomial Principal Component Analysis (MPCA) (Buntine, 2002), probabilistic Latent Semantic Indexing (PLSI) (Hofmann, 1999a,b, 2001, Papadimitriou, Tamaki, Raghavan, and Vempala, 1998), and Latent Dirichlet Allocation (LDA) (Blei, Ng, and Jordan, 2003). There has been some research on expressing these decompositions in a unified way ((Buntine and Jakulin, 2006) and (Singh and Gordon, 2008)).

2.3 User Interest Profile For Personalized Web Search - Non Folksonomy based

Google's innovative page ranking search (Brin and Page, 1998) revolutionized the use of SEs. PageRank uses the citation graph of the Web along with the introduction of link analysis in SE systems. SEs, such as Google, Yahoo, and MSN, do a commendable job for experienced users, but fail to satisfy the needs of naive users. (Teevan, Dumais, and Horvitz, 2007) reported that although SEs provide the best possible result set, they are not satisfactory at individual user levels. Search results can be improved by personalization (Chirita, Firan, and Nejdl, 2006, Eugene, Eric, Susan, and Robert, 2006, Kelly and Teevan, 2003, Ma, Pant, and Sheng, 2007, Shen, Tan, and Zhai, 2005, Teevan, Dumais, and Horvitz, 2005), by, for example, recommending varying results to different users for the same query. The results are differentiated based on user interests, which are obtained from the user's *UIP*. Automatic construction of a *UIP* usually deals with the observation of user browsing behaviour. (Kelly and Teevan, 2003) reviewed several possible approaches to inferring user preferences by categorizing user behaviour across many dimensions such as examine, retain, and reference. (Agichtein, Brill, and Dumais, 2006) organized user interests as a set of features that are organized into three groups: Query-text, Click-through, and Browsing. The Query-text feature includes result title, URL, and summary. The Click-through feature includes Click-Frequency (number of clicks for the result), IsClickBelow (whether there was a click on a result below

2.3 User Interest Profile For Personalized Web Search - Non Folksonomy based

the current URL), and IsClickAbove (whether there was a click on a result above the current URL). The Browsing feature includes TimeOnPage, TimeOnDomain, and the deviation of the dwell time from the expected dwell time for a query. (Shen, Tan, and Zhai, 2005) collected user interests from clicked document summaries, titles, Click-Through histories, and query histories that were accumulated over a session. (Teevan, Dumais, and Horvitz, 2005) and (Chirita, Firan, and Nejdl, 2006) used the files on the user's desktop to construct a *UIP*. A major limitation of these approaches is that there can be a lot of terms on the user's desktop, which makes a *UIP* noisy or misleading. (Das, Datar, Garg, and Rajaram, 2007) used collaborative filtering (CF) for personalization. The underlying assumption of the CF approach is that users who agreed in the past tend to agree again in the future.

A rather simplistic approach to construct a *UIP* is to explicitly ask a user for his/her topics of interest. The *UIP* is then used for filtering search results by checking content similarity between the returned Web documents and the *UIP*. Early versions of Google personalization asked the user to choose the categories of interest. The Google SE applied this information to filter search results. An inherent limitation of this approach is that user interests are subject to changes over time. Moreover, (Carroll and Rosson, 1987) showed that users are reluctant to provide explicit information about their interests or any explicit feedback on search results. Other important methods using ontologies emerged as well in which a *UIP* is constructed by classifying Web pages in the user's web browser cache into appropriate concepts in the reference ontology

2.4 User Interest Profile for Personalized Web Search - Folksonomy based

(Gauch, Chaffee, and Pretschner, 2003) and (Speretta and Gauch, 2005a) or ODP (Chirita, Firan, and Nejd, 2006).

2.4 User Interest Profile for Personalized Web Search - Folksonomy based

Recently, some research works have investigated social bookmarking services for building and applying a *UIP* for personalized search (David, Iván, and Joemon, 2010, Kumar and Kim, 2012, 2011, Noll and Meinel, 2007, Xu, Bao, Fei, Su, and Yu, 2008) and resource recommendation (Abel, Gao, Houben, and Tao, 2011, Andriy, Jonathan, Bamshad, and Robin, 2008, Fabian, Nicola, Eelco, and Daniel, 2010, Vallet and Castells, 2012).

The approaches by (Noll and Meinel, 2007), (Xu, Bao, Fei, Su, and Yu, 2008), and (David, Iván, and Joemon, 2010) for personalized search builds a *UIP* from the tags that the user uses to annotate resources. A Resource Profile(*RP*) for a resource is constructed from the tags that the community has used to annotate it. A resource clicked by a user manifests the user's interest in it and possibly the tags associated with it. Tags assigned by a user to a resource can hardly be a complete description of the resource. However, collective tagging of a resource by a community of users provides a more complete description of it. We believe that there are syntactical differences between the search terms that a user uses and the terms found in a search result document. Each user has a specific vocabulary of terms that he/she uses to formulate a query. And

2.4 User Interest Profile for Personalized Web Search - Folksonomy based

Table 2.1: A comparison summary of the proposed approaches with the other similar approaches that uses folksonomy for personalized search. (a)Source of terms for constructing a *UIP*, (b) Web document Representation, (c) Similarity Measure, (d)First-Order Co-occurrence, (e) Second-Order Co-occurrence, (f)Clustering of terms in a *UIP*, (g) *UIP* and resource length normalization factor

	<i>tfUIP</i> Noll and Meinel	<i>tfIdfUIP</i> Xu et al.	<i>tfIdfCUIP</i> Andriy et al.	<i>svdCUIP</i>	<i>modSvdCUIP</i>
(a)	User annotations	User annotations	User annotations	Annotations by the community	Annotations by the community
(b)	Resource Profile (folksonomy based)	Resource Profile (folksonomy based)	Resource Profile (folksonomy based)	document contents	document contents
(c)	dimensionless cosine similarity measure, Equation 2.1	tfIdf cosine similarity between a <i>UIP</i> and a <i>RP</i> , Equation 2.2	cosine similarity between a global cluster matching a user query and the <i>RP</i>	cosine similarity between the matching cluster in the <i>CUIP</i> to the user query and the document contents, Equation 4.6.	calculates the cosine similarity between the matching cluster in the <i>CUIP</i> to the user query and the document contents, Equation 4.6.
(d)	Yes	Yes	Yes	Yes	Yes
(e)	No	No	No	No	Yes
(f)	No	No	Yes	Yes	Yes
(g)	No	Yes	Yes	Yes	Yes

each author of a document has his/her own vocabulary of terms too. Chances are that the vocabularies are different. The rift effectively results in the low similarity score or re-ranking score between the search result and the *UIP*. Note also that there can exist similarity in semantics among the terms in the user's *UIP* and the *RP* of the result document. If a *UIP* consists of all the tags, used by a community of users, to annotate the resources of user interests, it is very likely to have a greater correspondence between the *UIP* and the *RPs* of result documents. Hence, it is our proposal that a *UIP* should consist of all the tags used by a community of users to annotate the documents or resources clicked by the user. We have adapted the approaches presented in (Noll and Meinel, 2007), (Andriy, Jonathan, Bamshad, and Robin, 2008), and (Xu, Bao, Fei, Su, and Yu, 2008) to construct a *UIP* by amalgamation of tags from the *RPs* of the resources or Web documents clicked by the user. We are of the opinion that any of these approaches can be benefited by the application of SVD, an approach proposed by us to construct a *CUIP*.

2.5 Personalized Search

One of the issues with personalized search is how to acquire the index? The construction of an index is a tedious process. An alternative option is to use the search results available from the SE. Most SEs do not allow scrapping of search results. However, they do provide search APIs with limited access and some restrictions. Researchers use Google API, Yahoo API, or MSN API to retrieve

search results. (Kumar and Kang, 2008) provided a comprehensive look at the differences in search results obtained from a SE and a SE API for the same input query, i.e., How well a SE API surrogates a SE? The following differences are reported: freshness, accuracy, ranking, the number of results, and the difference in index. They reported that Yahoo SE and Yahoo search API have same search quality, this is to say, underneath both use the same index, unlike Google API and MSN API use a different index than used by Google and MSN, respectively. This work uses Google API for retrieving search results.

(Pitkow, Schütze, Cass, Cooley, Turnbull, Edmonds, Adar, and Breuel, 2002) described two approaches to personalizing Web search results: query expansion (Chirita, Firan, and Nejd, 2006, Gauch, Chaffee, and Pretschner, 2003, Speretta and Gauch, 2005a) and re-ranking of search results (David, Iván, and Joemon, 2010, Ferragina and Gulli, 2005, Koshman, Spink, and Jansen, 2006, Noll and Meinel, 2007, Wang and Jin, 2010). In query expansion, user interests are conflated with a given query, and the expanded query is used for searching the Web. For re-ranking of search results, the SE results are re-ranked by computing the similarity between the document contents and the terms in the UIP.

(Agichtein, Brill, and Dumais, 2006) used supervised machine learning technique, named RankNet, for re-ranking search results. (Dou, Song, and Wen, 2007) used S.E logs for constructing user profiles. Further they re-rank search results by computing a personalized score for each URL in the result set. They introduced four formulas for re-ranking: two methods closely relate to collab-

orative filtering, and the other two relate to personal level. (Ferragina and Gulli, 2005) proposed web snippet clustering, in which the search results are presented hierarchically using web snippets. It clusters snippets returned by a SE into a hierarchy of folders which are labelled with variable length sentences. The labels are named such that they represent the theme of the snippets contained into their associated folders. For personalization, users can select a set of labels, and ask the SE to filter out all other labels except the selected ones. Note that their approach is bounded towards clustering search results, whereas our approach is bounded towards clustering terms to generate a CUIP and using the CUIP for personalized search.

The method by (Noll and Meinel, 2007), referred to as *tfUIP* in this thesis, re-ranks a document by computing the dimensionless cosine similarity between the tags in the *RP* of the document and the *UIP*.

$$tfUIP(UIP, d) = \sum_{t \in UIP, tf_d(t) > 0} tf_{UIP}(t) \quad (2.1)$$

The method by (Xu, Bao, Fei, Su, and Yu, 2008), referred to as *tfIdfUIP*, re-ranks a document by computing the cosine similarity between the tags in the *RP* of the document and the terms in the *UIP*.

$$tfIdfUIP(UIP, d) = \frac{\sum_t (tf_{UIP}(t) \cdot idf_{UIP}(t) \cdot tf_d(t) \cdot idf_d(t))}{\sqrt{\sum_t (tf_{UIP}(t) \times idf_{UIP}(t))^2} \cdot \sqrt{\sum_t (tf_d(t) \times idf_d(t))^2}} \quad (2.2)$$

The method by (David, Iván, and Joemon, 2010), an adapted approach of (Xu, Bao, Fei, Su, and Yu, 2008), referred to as *tf-iuf* in our work, excludes length normalization factors of the *UIP* and documents from the similarity score computation, and includes the inverse user frequency and inverse document frequency.

$$tf - iuf(UIP, d) = \sum_t (tf_{UIP}(t) \cdot iuf_{UIP}(t) \cdot tf_d(t) \cdot idf_d(t)) \quad (2.3)$$

The justification for exclusion of document length normalization factor is similar to that of *tfUIP* that using the document length normalization factor would penalize the score of popular documents. The reason for exclusion of *UIP* length normalization factor is that in all computations of similarity scores, the *UIP* length normalization factor is constant. Similar to *tfUIP*, *tfIdfUIP* and *tf-iuf* use all terms in the user’s *UIP* for computation of similarity scores to re-rank search result documents.

Recent work(Bouadjenek, Hacid, and Bouzeghoub, 2013a, Bouadjenek, Hacid, Bouzeghoub, and Vakali, 2013b) on folksonomy based personalized search builds a personal document representation (PSDR) in a social collaborative setting. Further, a ranking function is proposed to rank documents using PSDR.

The method by (Andriy, Jonathan, Bamshad, and Robin, 2008) presented a personalization algorithm for recommendation in folksonomies, referred to as *tfIdfCUIP* in our work, which relies on hierarchical tag clusters. Their approach clusters the entire tag space of a folksonomy system to obtain one common,

global cluster structure available to those users who are registered with the folksonomy system. This restrains the outreach of the approach. Further, they gauge user interest in each tag cluster based on the user usage of tags for resources' annotations. A set of matching clusters extracted from the overall clustered tag space makes up a *CUIP* to be used for personalized resource recommendation. And, both tf-idf and tf are used to compute the similarity score of resources and a *CUIP*.

Our proposed methods, for personalized search based on *svdCUIP* and *modSvdCUIP*, use a *UIP* length normalization factor during similarity score computation because the methods expand the user query with the tags from the matching cluster in the user *CUIP*, and compute the similarity score between the expanded query and the document contents. The *UIP* length normalization factor varies in accordance with queries because each query may match to a different tag cluster. Because *RPs* can only be constructed for a small subset of documents, we refrain from using *RPs* of documents for ranking them. The methods calculate the similarity between the expanded query and document contents. In fact, we have found that it is only possible to construct *RPs* for approximately 50% of Web documents when using social bookmarking services. This seriously jeopardizes the outreach or acceptability of personalized search systems.

In a nutshell, the *tfUIP* and *tfIdfUIP* re-rank the search result set by computing the similarity scores between the terms in the *UIP* and *RPs* of documents in the result set, whereas the proposed approaches are based on query expansion and use document contents for ranking search results.

2.6 Partnership Match

Existing work in the domain of Partnership Match is focused towards total ranking of the Suppliers (Chen, Lin, and Huang, 2006, Chen, Lee, and Wu, 2008, Dulmin and Mininno, 2003, Lin, Xu, and Xu, 2010, Liu and Hai, 2005, Sun, Ji, and Xu, 2009). Hence, these works provide some sort of weight procedure based on ANP (Chen, Lee, and Wu, 2008), Fuzzy Logic(Chen, Lin, and Huang, 2006), Data Mining(Lin, Xu, and Xu, 2010), and AHP (Liu and Hai, 2005). The research work related to Partnership Match can be classified into following categories: AI Systems (Chen, Lin, and Huang, 2006, Liu and Hai, 2005), Mathematical Models (Chen, Lee, and Wu, 2008, Choy and Lee, 2003, Dulmin and Mininno, 2003), Ontology Models(Li, Wu, and Yang, 2004a, Li, Huang, Liu, Gou, and Wu, 2001), Statistical(Petersen and Divitini, 2002), and Simulation Studies(Basnet and Leung, 2005, Cakravastia and Takahashi*, 2004). We place our work under Ontological models.

(Chen et al., 2006)propose to solve supplier selection or partnership establishment problem by building a hypothesis that there is an uncertainty involved in decision variables of partner attributes. Therefore, they propose to use fuzzy algorithms. However, their work is based on preliminary screening, which means, the process is partially automatic. Most of the research work in this area revolves around using Mathematical Models(Chen, Lin, and Huang, 2006, Choy and Lee, 2003, Dulmin and Mininno, 2003, Min, 1994). Some authors formulated the partnership establishment problem as Analytic Network

2.6 Partnership Match

Process(Bayazit, 2006, Chen, Lin, and Huang, 2006), some use Case Based Reasoning(Choy and Lee, 2003), and Multi Attribute Utility tool(Min, 1994, Sun, Ji, and Xu, 2009). Interestingly, an organization profile which consists of quantitative attributes and qualitative attributes has to be modeled such that they can be effectively used for numerical calculation(Dickson, 1966). The problem arises when modeling qualitative attributes for numerical calculation- solution to which is often provided by using mathematical Models. The qualitative features modeled use a scale indicating the strength with which one factor dominates another with respect to a higher level factor. However, in the aforementioned research work, the list of attributes to model a profile is not comprehensive, for ex: (Choy and Lee, 2003) and (Dulmin and Mininno, 2003) fail to take into account marketing capabilities, financial stabilities, and cultural alignment etc. We have tried to cover all the features for modeling a profile. This enforces the fact that different companies have different specific requirements concerning supplier evaluation. For instance, (Schmitz and Platts, 2004) used a semi-structured questionnaire in several European locations to collect opinions and suggestions from automobile suppliers on vendor performance evaluation. One of the key results of their study is that the evaluation of supplier includes management information, communication, motivation of suppliers, coordination and alignment, decision making and priority learning. A number of simulation studies with a focus on the partner establishment have also been published. (Crama et al., 2004) formulated a non-linear 0-1 programming problem with complex quantity discounts offered by different suppliers

2.6 Partnership Match

and alternative product recipes. (Cakravastia and Takahashi*, 2004) created a simulation model to determine which supplier to select for business and the volume assigned to each of those suppliers. Finally, (Basnet and Leung, 2005) created a simulation model to determine what products to order in which quantities from which supplier in which periods to satisfy a given demand stream. One major task of purchasing manager is selecting the right supplier. Suppliers have varied strengths and weakness which requires meticulous evaluation by the purchasing manager before ranking them. The foremost task is to establish the criteria or features for supplier evaluation. (Weber et al., 1991) classified 74 articles, on the 23 criteria from (Dickson, 1966), related to supplier selection and discussed the effect of various features on supplier selection. Since different enterprises have different requirements in terms of supplier evaluation, i.e., they use different set of features therefore in this work, we have arranged a comprehensive list of features required by purchase managers and further represented those features as an ontology. Ontologies are the structural frameworks for organizing information and are used in Grid Computing(Lee, Lee, Noh, and Han, 2010) (Jang, Lee, Noh and Han 2010), WWW (Sui and Zhao, 2009), systems engineering (Pham and Jung, 2010), etc, as a form of knowledge representation about the world or some part of it. (Li et al., 2001) and (Li et al., 2004a) use ontology for modeling partner profile; however, authors fail to provide any case study that can demonstrate their work. (Petersen and Divitini, 2002) use statistical model for calculating similarity between partners. Their model particularly works for software projects; hence the features for modelling

2.6 Partnership Match

profiles are more technically oriented rather than being generalized. It uses an agent oriented approach and Multi-Attribute Utility Function to determine the score of partners which is further used for ranking. This work suffers from the drawback that it only works for virtual enterprise that is formed for a software project.

3

Mining anchor text for building User Interest Profile: A non-folksonomy based personalized search

The very first search engine was developed by Gerard Salton, and it was called the SMART information retrieval system(Salton, 1971). The first pre-web search engine was Archie(Van Couvering, 2008), which allowed searching for file names of a database. The early search engines retrieved results from their indexed database and displayed the cached pages based on keyword match and similarity measures. Traditional indexing methods worked quite well for database or structured information but later it was discovered that they are not

compatible for indexing unstructured information such as World Wide Web. The search engines based on simple indexing technologies were Lycos, Alta Vista etc. (Brin and Page, 1998) proposed an innovative page ranking system which revolutionized the use of search engines. Page rank uses the citation graph of the web and Google introduced link analysis in the search engine systems.

To improve the quality of search results returned by a search engine, many solutions have been proposed: first is to use a Vertical Search Engine (Koshman, Spink, and Jansen, 2006) for specific information needs, second is the use of a personalized search engine (Chirita, Firan, and Nejd, 2006, Das, Datar, Garg, and Rajaram, 2007, Ferragina and Gulli, 2005, Gauch, Chaffee, and Pretschner, 2003, Speretta and Gauch, 2005b, Sun, Zeng, Liu, Lu, and Chen, 2005, Teevan, Dumais, and Horvitz, 2005, 2007), and third is to improve search engine results (Chakrabarti, Dom, Gibson, Kumar, Raghavan, Rajagopalan, and Tomkins, 1998a, Chakrabarti, Dom, Raghavan, Rajagopalan, Gibson, and Kleinberg, 1998b, Haveliwala, 2002). Personalized Search has emerged as an effective solution to improve quality of search results. Using Topic Distillation (Chakrabarti, Dom, Gibson, Kumar, Raghavan, Rajagopalan, and Tomkins, 1998a) and ARC (Chakrabarti, Dom, Raghavan, Rajagopalan, Gibson, and Kleinberg, 1998b), Chakrabarti et al. has showed how quality of search results can be improved. Another similar attempt by Haveliwala (2002) used hub vectors limited to 16 for calculation of topic sensitive page rank. These approaches will improve search results but they do not provide different results for different

users. Using a Vertical Search Engine is not appropriate in all cases as they have an inherent restriction that they are restricted to one specific domain.

I want to leverage upon feature based user profiling, refer Figure 1.1, for building a profile of user interests. In this chapter, a profile of user interests is built from the anchor text of the clicked Web pages in the user search history. This type of method is called as non-folksonomy based method for building a profile of user interests. The anchor text represents the feature that is being mined to represent user interests. In chapter 5, I propose a more advanced method to build a profile of user interests that uses a different feature. Both non-folksonomy and folksonomy based methods in chapter 4 and 5 are used for personalized search. In chapter 6, a practical approach to build a use profile from explicit user involvement is presented, and the user profiles are used for partnership match.

This chapter makes the following contributions:

1. I propose a non-folksonomy based personalized search method, Exclusively Yours', that capitalizes on the anchor text to construct a User Interest Profile (*UIP*).
2. I propose a term-weighting method specifically targeted to this work with the goal of accumulating weight of terms emanating from the linked Web pages of clicked documents.
3. I also propose a model to logically segregate a *UIP* into two parts based on the latency of terms in the *UIP*. It effectively discounts term weight

of those terms in the *UIP* that have not been updated over a period of time.

4. The proposed method is compared with the other non-folksonomy based personalized search methods and with the non-personalized Web search.

To achieve good personalization (Ferragina and Gulli, 2005), three requirements have been stated: full adaptivity to the changing user behaviours/needs, privacy protection, and scalability. Our proposed method satisfies all the three aforementioned requirements. To take care of user behaviour needs that may change over a period of time, we construct two types of user profile, p_{perm} and p_{temp} . Regarding the privacy, we make no attempt to infringe in user personal data or personal files as has been done by few personalization techniques (Chirita, Firan, and Nejdl, 2006, Teevan, Dumais, and Horvitz, 2005). Regarding scalability, we tested our system for many months and with many users, the results obtained were satisfactory.

3.1 Exclusively Yours'

Figure 3.1 is an illustration of Exclusively Yours' system architecture that provides personalized search results for a given user query. On the client side, a user requests a query and chooses a search engine from the available four options (Google, Yahoo, MSN, and Naver). The retrieved search results (a set of ranked URLs) are logged along with the query and the user ID. Each user is supposed to register before he/she can use the proposed system. Each

3.1 Exclusively Yours'

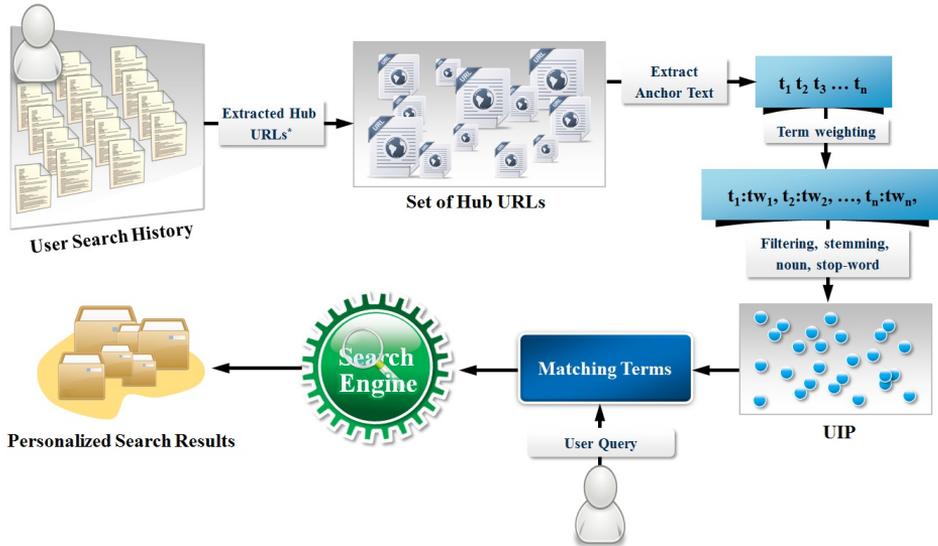


Figure 3.1: System Architecture of Exclusively Yours'

user logs in using his unique user ID. The user ID and other information are logged. The logged information is used during experiment for identification of a session. If a user clicks or downloads a URL, the system logs the selected URL along with the query and the user ID. The anchor text extraction module extracts anchor text and its surrounding text from the associated hubs of each URL clicked or downloaded by the user. We have proposed a weighting scheme that assigns weight to each extracted term. The weight is computed in the weight computation module. The weight assigned is based on the rank of URL and the rank of associated hub URL that contains the anchor text. Moreover, the extracted terms are stored in an indexed file along with their weights, and

various other attributes. The User Interest Profile (*UIP*) consists of extracted terms which will be used later for expanding user query.

3.1.1 Infer User Interests

This section describes our approach and the experiments that we use to set values for the small number of parameters in the algorithm. We have divided the whole process into three phases: 'training' phase, 'weighting' phase, and 'testing' phase. Given a query q , let U be the set of URLs returned from a user selected search engine which can be Google, Yahoo, or Naver (a Korean Search Engine). Let V ($V \subset U$) is the set of URLs clicked or downloaded by the user as shown in Figure 3.2. We now propose two fundamental ideas. The first idea,

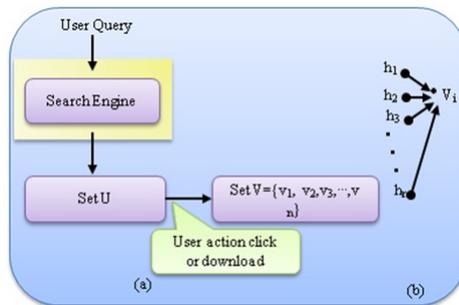


Figure 3.2: Set U represents URLs returned by a search engine and set V represents URLs clicked or downloaded by the user. On the right (b), URLs $h_1, h_2, h_3, \dots, h_n$ are hub URLs for URL V_i .

we need to find user interests using hyperlink structure. The second idea which is explained in detail in section 3.1.2 expands user query by conflating it with

a set of related terms from the *UIP*. To achieve the goal of first idea, hub pages are determined for the web pages in set V : for each URL v_i in the set V , find the top n web pages that are hub pages of v_i , i.e., web pages that have a link to a page v_i , as shown in Figure 3.2. If a page u has a link to a page v then u is a hub page for a page v .

We believe that the URLs that a user clicks or downloads are related to his/her interests. It has been reported by (Kraft and Zien, 2004) that there exists similarity between search queries and anchor text. They also showed that anchor text is a succinct description of a web page. Therefore we extract anchor text and its surrounding text from the hub pages of URLs clicked or downloaded by the user to create an index file of extracted terms.

We are only interested in hub pages because it gives a comprehensive description of hyper-linked outgoing linked web pages. From each of the n hub pages corresponding to v_i , extract a window of size 50 bytes surrounding the anchor texts from an anchor tag that has a href (hypertext reference) link to page v_i . A similar work by McBryan (1994) has defined a window of size 50 bytes surrounding an anchor text as anchor window. To extract the text circumscribing the anchor text, the first step is to get rid of html tags around it. The following step is removal of stop words and stemming. The resulting text is indexed and assigned weight w_i . The process of calculating weight is explained in the next section.

Here is an example to demonstrate how anchor text and its surrounding text is extracted. For ex: a user entered a query *Hollands Opus*. The topmost result

Table 3.1: Top three Hub URLs for the IMDB URL

www.imdb.com/title/tt0113862/	
$HubURL_1$	www.math.harvard.edu/~knill/mathmovies/index.html
$HubURL_2$	www.salocin.com/weblog/archives/2004_04.html
$HubURL_3$	http://www.timlebon.com/wise-books/

Table 3.2: Terms extracted from the $HubURL_1$

www.math.harvard.edu/~knill/mathmovies/index.html				
Force	Choose	mozart	read	write
Long	Division	Cut	Art	Kid
Holland's Opus	Movie	Math		

is a URL (<http://www.imdb.com/title/tt0113862/>). This URL is provided by IMDB and provides the comprehensive information about the movie *Holland's Opus*. If the user clicks this URL, the hub URLs are extracted using the query (`link:http://www.imdb.com/title/tt0113862/`) directed to yahoo web service. As a result, the top three URLs that point to IMDB URL returned by yahoo service are presented in Table 3.1. The $HubURL_1$ has an anchor text that has HREF link to IMDB URL. On careful examination of this anchor text, we find out that most of the text surrounding the HREF was `<table>` tags. After removal of tags, the extracted text is *If I'm forced to choose f... to read and write about*. Interested readers can find the complete text by browsing $HubURL_1$. Table 3.2, shows terms returned after parsing, stop-word removal, and stemming. This is the final set of terms which is indexed with weight assigned to each one of them. Extracted terms from the k hub pages are indexed in a file called as index file. Each term in the index file is assigned

a weight. The procedure for weight calculation is explained in the next section. We believe that the text around HREF links to a page v_i is descriptive of the contents of v_i .

3.1.2 Weight Computation

One of our major contribution is the computation of the weight w_i for each extracted term. The idea is to assign the log of rank of the hub page that contains the anchor text to w_i as shown in equation 3.1 where R_{kj} is the rank of k^{th} hub page associated with the j^{th} URL clicked/downloaded by the user. Note that, R_{kj} is subtracted from the count of results in the first page.

$$w_i = \sum_{j=1}^{|V|} \sum_{k=1}^{|H|} \frac{\log R_{kj}}{\log R_j} \quad (3.1)$$

The denominator, R_j is the rank of j^{th} URL clicked by the user, acts as a parameter of penalization. It controls how much a rank at a lower position is penalized. Because $\log 1 = 0$, which will result equation 3.1 to infinity, instead we have used $\log 1 = 1$ for computation.

The parameter H represents a set of hub pages associated with the URL j . The double summation in equation 3.1 accumulate term weights if a term reoccurs in either or all the hub pages associated with the URL j . Further, if an extracted term appears in a web page that already exists in an index file, then its weight is cumulatively added. Also note that, the value of weight w_i is highly responsible for separating noise, i.e., those terms which do not correspond to user interest

will not occur too often and hence will have lower weight. Whereas the terms the occur too often will subsequently have higher weight thus indicating user interests. It can be argued that there will be a lot of such terms. We found out that there is indeed a lot of terms that represented user interests; these terms were also somehow related, for ex, from Table 3.1, one can see that movies, art, Mozart are closely related terms, they have high contextual similarity. To resolve the ambiguity, such contextually similar terms can be grouped together, i.e., those terms that are contextually similar are grouped together. One term which has highest weight can collectively represent such a group of terms. To determine the contextually similarity between terms, we have used Normalized Information Distance (NID)(Li, Chen, Li, Ma, and Vitányi, 2004b). The idea behind NID is that the terms that are closely related occur together in almost all the documents and hence their NID value evaluates to close to 1. For ex: if terms t_1 and t_2 are closely related, then the number of documents in which t_1 appears will be more or less similar to the number of documents in which t_2 appears. Those terms that are not closely related, have less frequency to occur together and their NID value is a larger number. Since user interests may change over a period of time, a *UIP* is logically viewed in two forms, p_{perm} and p_{temp} . The p_{perm} represents *UIP* for all days prior to current day and p_{temp} represents *UIP* for the current day. The *UIP* p_{temp} consists of terms collected for the current day and p_{perm} consists of terms collected during few days before current day. p_{temp} is constructed through the following process. We construct a vector a_t of terms collected from the hub pages corresponding to each web

page in V as follows:

$$a_t^i = \{a_{t_1}^i, a_{t_2}^i, a_{t_3}^i, \dots, a_{t_n}^i\} \quad (3.2)$$

where n is the number of terms extracted from URL v_i and its corresponding hub pages. The term at collected during one session is calculated as follows

$$a_t = \bigcup_{i=1}^{|V|} a_t^i \quad (3.3)$$

We divide user activity into various sessions during a particular day, i.e., each query represents one session. Moreover, we take union of all terms collected over all the m sessions in a single day which is represented as profile p as shown below.

$$p = \bigcup_{j=1}^m a_t^j \quad (3.4)$$

Finally, each term t_i is associated with two attributes; weight w_i and date of activity $a(t)$. The term date of activity is defined as the date when the weight of term was last updated. As shown in the equation above, a *UIP* is a collection of terms. The second idea which is presented in Section 3.1.2: expands user query by conflating the closing related terms in a *UIP* with the user query. The expanded query is submitted to a search engine which returns a set of URLs that are presented to the user.

3.1.3 Query Expansion

Query expansion represents the testing phase. In this phase the query terms entered by the user is expanded with the top k terms which were collected in training phase. The top k terms are determined by calculating the contextual similarity of terms in the *UIP* and user query terms. The contextual similarity is calculated using NID (Li, Chen, Li, Ma, and Vitányi, 2004b) as explained in the previous section. The weight w_i is used for identifying the most relevant user interests and its application is described below. After extracting the contextually similar and closest term to user query, we divide the weight w_i of each term a_i in profile p with the exponential over difference of current date and date of activity as shown in equation 3.5. The date of activity is used to maintain the validity of profile

$$P_{temp} = p$$

$$P_{perm} = \frac{p}{e^{c(t)-a(t)}} \quad (3.5)$$

where $c(t)$ is current date and $a(t)$ is date of activity. The division operation reduces the importance of a profile as it gets older. Thus, it takes care of changing user interest. Note that, if a profile consists of some terms that got updated recently, their weight increases and also their date of activity changes to the most recent one. In other words, a collection of terms which

3.2 Exclusively Yours' Algorithm

got introduced long time back and has not been updated lately, means it no longer reflects user interest. The final profile P_{final} can be calculated as shown in equation (15), which is a union of P_{temp} and P_{perm} .

$$P_{final} = P_{temp} \cup P_{perm} \quad (3.6)$$

The expanded query is submitted to a search engine of user choice. We decided to choose the value of k as four i.e. conflate the top four or less contextually similar terms with the user query. (Phelps and Wilensky, 2000) reported in their research that five terms are sufficient to determine web resource uniquely.

3.2 Exclusively Yours' Algorithm

In this section, we briefly explain about the web search APIs used and give an overview of the algorithm behind the proposed approach. Figure 3.3 presents a snapshot of Exclusively Yours' user interface. All the three web search APIs provide the same type of functionality. We can use web search APIs to request query, receive total number of results, URLs, snippets, and title. Although the APIs are provided for free, they impose certain restrictions like the number of query terms, the number of queries that can be issued in one day, and the number of results in one set. Google and Yahoo return 10 results in one set, whereas Naver returns all the results as an xml file. We developed Exclusively Yours' using Java technologies, HTML Tidy, DOM API¹, and Apache

¹<http://tidy.sourceforge.net>

3.2 Exclusively Yours' Algorithm

Server. The user is expected to login and choose a particular search engine before requesting a query. Individual user information such as query submitted, results returned (snippets and titles), total number of results, and web pages clicked by the user are logged in the database which is used later for experiments. Using web search APIs has many advantages: The system is dynamic, personalization is based on data readily available to the search engine, and we don't need to invade user personal information. Following is a brief description of Exclusively Users' algorithm. The algorithm itself doesn't deserve

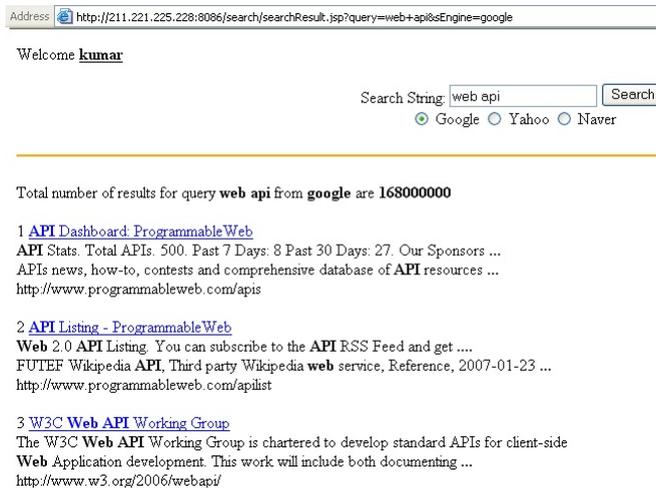


Figure 3.3: A Snapshot of Exclusively Yours' user interface

much explanation as it has already been explained in previous sections. In brief, procedure PERSONALIZE-RESULT forms the core part of Exclusively Yours' system. It primarily does two jobs: (1) creates a profile using procedure CREATE-PROFILE, (2) extracts anchor text along with its surrounding text

3.2 Exclusively Yours' Algorithm

using EXTRACT-ANCHOR Procedure. Moreover, it conflates user profile with the query terms and observes user browsing behaviour, i.e. URLs clicked by the user. The procedure CREATE-PROFILE creates user profile using terms, their weights and activity date stored in index file. The procedure EXTRACT-ANCHOR saves terms along with their weights and activity date extracted from the hub URLs of clicked web pages. Figure 3.4 presents a snippet of code that receives and presents the user with URLs, snippet, title, and the total number of results for user query qs. The first if condition investigates, if user selected yahoo search engine, in that case, it creates an instance of YahooBean. The method setDirectiveArg() sends the user query to the yahoo server. The first 10 URLs, snippets and title are returned using the method, getResult(), getVectorSnippet(), and getVectorTitle() respectively. The method getTotalNumberOfResults() returns the total number of results returned by the search engine. Finally, the for-loop presents URL, snippet, and title to the user. The displayed 10 results are logged in the database. On clicking any of the URL, displayURL.jsp executes, which updates the record pertaining to URL clicked and redirects the browser to the appropriate URL. Further, anchor text extraction module executes to extract the anchor text and its surrounding text from the hub pages of clicked URL as shown in Figure 3.4. We developed a class HTMLParser that takes two inputs, the hub URL and the URL of clicked page. This code starts with the creation of an object of type HTMLParser. Finally the method extractAnchorText() of HTMLParser uses HTML Tidy to fix mistakes if any in the hub URL. After fixing the hub URL, it uses DOM

3.3 Experiments

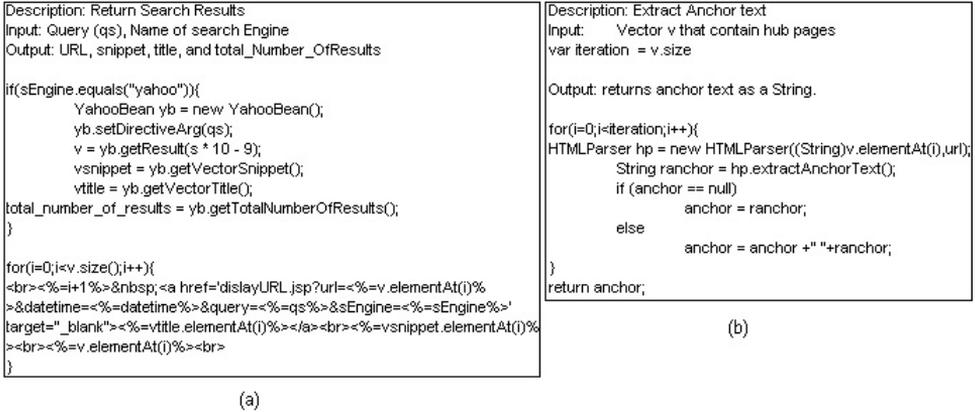


Figure 3.4: (a) Display URLs, snippet and title (b) extracts anchor text and its surrounding text from hub URLs.

API to extract anchor text and surrounding text from the hyperlink that links to URL clicked by the user.

3.3 Experiments

The objective of query expansion is to improve the precision of returned web search results. Hence, we evaluate our system over a large set of queries. We use two measurements to compare the performance of Exclusively Yours' personalized web search system with the original search engine: Average Rank and Discounted Cumulative Gain. To demonstrate the effectiveness of Exclusively Yours', Section 3.3.1 presents the parameter and data sets used for experiment. The metrics used for evaluation are described in Section 3.3.2.

Section 3.3.3 reports the comparison of Exclusively Yours' with some of the closely related personalized approaches, and section 3.3.4 compares Exclusives Yours with non-personalized search engines.

3.3.1 DataSet

In this section, we demonstrate the status of our system as it passes through various phases and the output thereafter. For the experiment purpose, our system was used by 15 volunteers over a period of one month. The 15 volunteers were students, professors, and researchers from various departments at Inha University and Suwon University in Korea. To test the full capability of our system we deliberately selected three volunteers from different departments such as Computer Science, Metallurgy, Biology, History, and Chemistry. In the span of one month, we collected approximately 2450 queries. The first fifteen days correspond to training phase, and the rest of fifteen days correspond to testing phase. Our system learns user behaviour and construct index file of extracted terms in the first 15 days. For the rest of 15 days, it does both the things; construct index file, update user profile and return personalized results. The number of days selected for training phase is purely empirical. We observed that a user needs at least 50-65 queries over a period of one week such that the proposed system can infer his/her interests. Just to make sure that a user inputs 50-65 queries, we assigned a period of 15 days for training. Apart from that, if a user thinks that he is searching something which is unconventional and should not be observed, he can choose to switch off the personalized system

and use the search results from the original search engine. In that case, our system neither extract terms nor does query expansion. Finally, we have tested all our results for test of significance (t-Test). The test condition is whether the personalized search result set improves the search quality when compared with the search result set of non-personalized search engine.

3.3.2 Evaluation Metrics

The metric Average Rank Manning et al. (2008) is used for measuring the quality of personalized search. The Average Rank (AR) of a query q is defined as shown in equation 3.7.

$$AR_q = \frac{1}{|V|} \sum_{p \in V} R(p) \quad (3.7)$$

where $R(p)$ is the rank of URL p . The final AR over all the queries for a use is computed as shown in equation 3.8. Smaller value of AR indicates better placement of results.

$$AR = \frac{1}{|Q|} \sum_{q \in Q} AR_q \quad (3.8)$$

Second metric that we used for measuring the quality of results is Cumulative Gain Järvelin and Kekäläinen (2002). A higher value of Gain Vector symbolizes more relevant results and vice versa. For example: if the highest value of CG is 20 in scenario1 and 12 in scenario2, that implies scenario1 has more highly relevant or relevant results as compared to scenario 2. The Cumulative Gain

Vector is calculated as shown in equation 3.9.

$$CG = \begin{cases} G(1) & \text{if } i = 1 \\ CG(i-1) + G(i) & \text{otherwise} \end{cases} \quad (3.9)$$

Third metric used for measuring the ranking quality is Discounted Cumulative Gain (*DCG*). Järvelin and Kekäläinen (2002). *DCG* is particularly useful when results are evaluated at difference relevance levels (highly relevant, relevant, and not relevant) by assigning them difference gain values. The idea behind *DCG* is, the greater the rank, the less accessible that URL is and hence less valuable it is to the user. The equation 3.10 shows the formulae used for computation of *DCG*.

$$DCG = \begin{cases} G(1) & \text{if } i = 1 \\ DCG(i-1) + \frac{G(i)}{\log_b(i)} & \text{otherwise} \end{cases} \quad (3.10)$$

For the purpose of this experiment, we have used three different relevance level $G(i)=2$ for highly relevant results and $G(i)=1$ for relevant results and $G(i)=0$ for not relevant results. Also b is the parameter of penalization; we have taken value 2 for b .

3.3.3 User Profile Efficacy

The objective of this section is to demonstrate the effectiveness of our proposed personalization method when compared with similar personalized search methods. We constructed *UIPs* using different methods: anchor text and its

3.3 Experiments

surrounded text (referred as anchor text), title, meta-tag keywords, and user browsing history. Note that, a user browsing history is available through the browser cache or using JavaScript's history object. To construct a *UIP* using title, we extracted title from the clicked URLs. To construct a *UIP* using meta-tag keywords, we extracted meta-tags from the clicked URLs. We were able to extract approximately 1050 browsed URLs from the browse cache. The $P@10$ was used as a performance measure which is shown in 3.5 which depicts that both anchor text with its surrounding terms and browser cache have almost same performance whereas user profile constructed using title of web page gives least performance. The reason for approximately similar performance of user profile constructed from anchor text along with its surrounding text and browser cache can be because both of them primarily represent extraction of anchor text from URLs. The difference lies primarily with the source of URLs. In the first case, i.e. anchor text user-based profile, the anchor text along with its surrounding text is extracted from the clicked URLs. Whereas in the second case, i.e. browser cache user-based profile, the anchor text along with its surrounding text is extracted from all the URLs that have been accessed by the user and are stored in cache. The browser cache based user profile can be thought of as it encompasses the URLs that were clicked by the user added with other URLs browsed by the user. Note that there is a slight drop in the performance of browser cache based user profile. It is because of some noise in user profile that gets induced due to URLs that were not clicked by the user but typed in the browser and browsed for some general information. Or

3.3 Experiments

they may be some pop-ups. The lower performance of title-based user profile and meta-tag keywords can be explained by the way a developer develop web pages. Web developers deliberately scribble such types of title and meta-tag keywords that are misleading and are not related to the content of that web page. Since, the efficacy of user profile using our method is significantly better than the other similar methods and is quite close to user profile constructed using browser-cache, we refrained going on with further experiments i.e. *DCG* and *NDCG*.

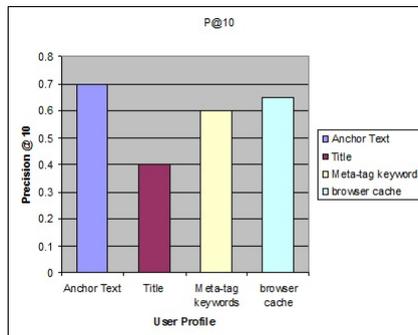


Figure 3.5: Efficacy of *UIP* constructed using different methods

3.3.4 Personalized vs. Non-Personalized Results

We shall now evaluate how the rankings of a non-personalized search engine and a personalized search engine differ based on the valuation we collected from our volunteers. We report two types of results here: one shows the results for an individual user and the other for a group. We found that, the personalized

search engine returned more relevant results as compared to results returned from a non-personalized search engine. However, the same query when issued by multiple users, received differed result sets and also the user’s rating was better. Figure 3.6 presents the CG curve for rank 1-30; the plotted graph com-

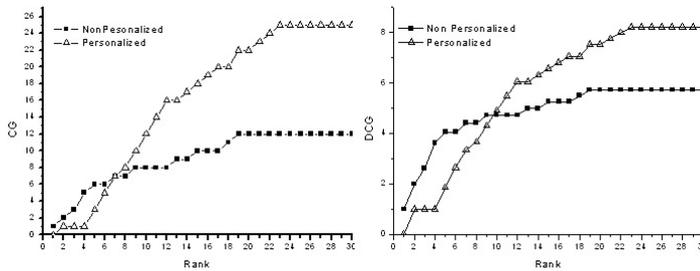


Figure 3.6: (a) Cumulative Gain (CG) Curve for an individual user query (b) Discounted Cumulative Gain (DCG) for an individual user query.

pares the user’s evaluation between non-personalized search engine results and personalized search engine results. Note that, the CG of a non-personalized search engine is flat at some places which indicates non relevant results. The steeper the curve, the more highly relevant results and the flat curve indicates not relevant results. The CG curve of a non-personalized search engine trails a horizontal line at rank 19 and onwards. This means, all of the relevant documents were available until rank 19. On the other hand, personalized search engine rank goes horizontal after rank 25. Moreover, the personalized search engine plot is steeper as compared to non-personalized search engine plot. Another metric that is worth noticing is the value of CG . The highest value of CG

for non-personalized search engine is 19, whereas for the personalized search engine, it is 25. The higher value for personalized search engine shows that more relevant results were presented to the user at higher ranks.

Figure 3.6(b) shows the *DCG* curves for ranks 1-30, that compares a non-personalized search engine results with the personalized search engine results. The \log_2 of the document rank is used as the discounting factor for the computation of *DCG*. One important thing to notice in this plot is the *DCG* of first 10 results. The *DCG* of initial results for personalized search engine is a little bit lower than the original non-personalized search engine results. There were a few such cases that this kind of situation occurred. However, from the plot for average *DCG* as shown in Figure 3.7(a) and the plot for average *CG* as shown in Figure 3.7(b), it is evident that results returned by personalized search engine have higher *DCG* thus representing better result quality. We

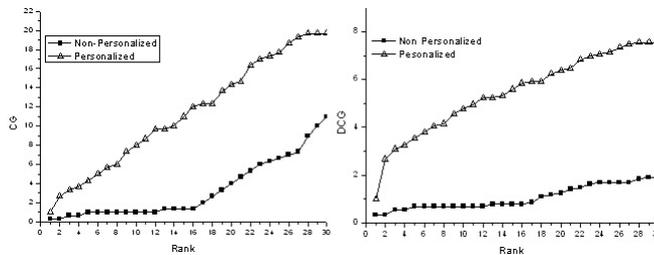


Figure 3.7: Average Discounted Cumulative Gain (DCG) Curve and (b) Average Cumulative Gain (CG)

investigated the reason for such discrepancy. The explanation follows. Our results are based on user interest and not based on query intent. We have been

3.3 Experiments

able to derive user interest and expanded the same with the user query but still there is a need to derive the intent behind the query. It will be an interesting future work to learn how to derive query intent and what effect does it have on search quality. Figure 3.8(a) shows the Average Rank (AR) for 5 departments

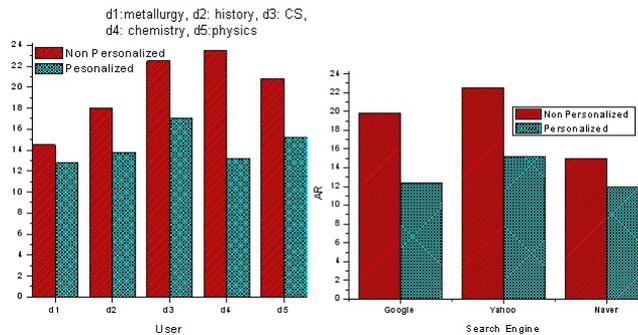


Figure 3.8: (a) Average Rank vs. each department (b) Average Rank vs. Search Engine

(metallurgy, history, computer science, chemistry, physics). It is clearly apparent that overall improvement is 37.6%. The best results are obtained for the chemistry department with improvement of 43.7%. We also learnt through our experiments that the need of personalization varies from query to query. As a matter of fact in some queries personalization produced bad results. For example, a user requested the query rank aggregation: a non-personalized search engine returned highly relevant results. The same query when conflated with his interests resulted in an increase in AR, which means bad quality results. We observed that in 5% of all the queries, the results returned lead to increase in AR. In another case, a user from physics department requested the query CNT

(Carbon Nano Tube), the top 17 results returned by a non-personalized search engine were all irrelevant, and hence in that case, there was significant improvement with personalization. This is another pointer where some improvement is required. We argue that if a system can distinguish between queries that require personalization and those that don't, then one can choose when and when not to apply personalization. This is an interesting future work, which we wish to carry on. Finally, Figure 3.8(b) shows the AR improvement when our personalized system is compared with non-personalized search engine. Over the entire experiment, our personalized system improved our Google, Yahoo, and Naver by 37.6%, 32.4%, and 20% (significant level with $p < 0.05$), respectively.

3.4 Conclusions

In this chapter, we proposed a personalized search method, Exclusively Yours', that infers user interests from user click through behavior. The URLs that a user clicks or downloads is used for the construction of *UIP*. Further, we extract the anchor text and its surrounding text from the associated hub pages of the URLs clicked by the user. In order to use extracted terms later for query expansion, we quantify the importance of each term by assigning a weight. We evaluated our personalized system with Google, Yahoo, and Naver using Cumulative Gain (*CG*), Discounted Cumulative Gain (*DCG*), and Average Rank (*AR*). We found that the proposed approach had significant improvement over non-personalized search engine except for 5% of the queries where personaliza-

tion had a negative impact. The average AR improvement is reported to be 30%.

We also observed that a *UIP* built from anchor text generates a better quality of search results resulting in user satisfaction, nonetheless, it has its own limitations. Anchor text was also found to contain some noise in the form of terms, such as 'next', 'go to', 'click here', etc. However, these anchor text was added without any maligned intention, unlike meta-tag keywords that contained terms not related to the Web document contents and were deliberately added to increasing the ranking of Web document. To further improve the quality of *UIP*, we propose a method that constructs a *UIP* from the tag annotations to the user clicked documents. Tags are annotated to a document by a wide variety of users, it is non-maligned, has rich content, and therefore we believe that it will result in a more enriched *UIP*. Personalization search methods that use tag annotations from a folksonomy system are termed as folksonomy based personalized search method.

4

Matrix factorization for building Clustered User Interest Profile: A folksonomy based personalized search

Quick ways to summarize documents, low latency to access documents, and convenient mechanisms to sharing them are all part and parcel of our daily lives. There is indeed a very large number of documents to deal with¹. Naturally, everyone will benefit if there exist smart programs to manage document

¹<http://googleblog.blogspot.in/2008/07/we-knew-web-was-big.html>

collection, tag them automatically, and make them searchable by keywords. To satisfy such needs, the multimedia, information retrieval, and computer vision communities have, time and again, attempted automatic document annotation, as we have witnessed in the recent past (Uren, Cimiano, Iria, Handschuh, Vargas-Vera, Motta, and Ciravegna, 2006). While many interesting ideas have emerged, not much attention has been paid to the direct use of automatic annotation for document search. Usually, it is assumed that good annotation implies quality document search.

One way of annotation that was widely discussed in the research community is the Social Semantic Web. It largely depends on pre-conceived ontology. However, due to a large amount of initial efforts demanded from web developer community, it did not achieve its success as was expected unlike Web documents which were/are hugely successful in realizing the current Web. Second impediment is that there is huge learning curve associated with Semantic Web. Unlike HTML where a layman can get started with building an HTML document after a couple of hours. Getting to grips with RDF/XML, SPARQL, and the other core technologies is a big ask for most developers. To then get useful semantic web applications out of these takes a couple more exhausting jumps of complexity, for instance, SWOOGLE - a semantic search engine, has reported that about one-third of the RDF files that it has harvested contains errors (Ding, Finin, Joshi, Pan, Cost, Peng, Reddivari, Doshi, and Sachs, 2004). Social Web has emerged as a hope that stands between the conventional Web and the Social Semantic Web. It stands for the culture of participation and

collaboration on the Web. Structures emerge from social interactions: social tagging enables a community of users to assign freely chosen keywords to Web resources. The structure that evolves from social tagging is called folksonomy and recent research have shown the exploitation of folksonomy structures is beneficial to information access.

In the previous chapter, Chapter 3, I proposed a non-folksonomy based method for personalized search that builds a *UIP* from the model proposed in Figure 1.1. The anchor text was used as a feature that is modelled as a user interest, and it was extracted from the hub pages of the clicked Web documents in the user search history. In this chapter, I propose another feature based approach to user profiling that first builds a *UIP* from the tags annotated to documents clicked by the user. Further, the tags in the *UIP* are grouped together into meaningful clusters, a *CUIP*, as perceived by the user. For ex: if a *UIP* consists of following terms, $[java, programming, travel]$, then based on user inclinations a *CUIP* could be, $[[java, programming], travel]$. For the same *UIP*, another *CUIP* could be $[[java, travel], programming]$. The former *CUIP* represents the context of term *java* as *programming*, whereas, the later *CUIP* represents the context of term *java* as *travel*. To discover hidden semantics, matrix factorization techniques are used in this work. This is to say, the proposed methods in this chapter are also based on feature based user profiling, refer Figure 1.1, where feature is tag annotations to Web documents clicked by the user. A profile is further enriched by discovering hidden semantics in its *UIP*, such profile is called as *CUIP*.

This chapter makes the following contributions:

1. We propose two methods to build a *CUIP* for personalized search: one that uses Singular Value Decomposition (SVD) to generate *svdCUIP*, and the other a variation of SVD, *modSVD*, to generate a *modSvdCUIP*. A set of pairs of the form (t, tw) , where t is a tag and tw is the accumulated weight of the tag t , constitutes a User Interest Profile (*UIP*). A *CUIP* is defined as a set of term clusters, where each term cluster consists of semantically related tags of user interests and tag weights.
2. An automatic evaluation method is proposed to test the proposed methods with the baseline search and folksonomy based personalized search approaches.
3. We performed experiments to evaluate the proposed methods on two different data sets. The first data set, called custom data set, was created from the search histories of 12 volunteers. This data set was organized to establish the ground truth for the evaluation of clustering tendency and clustering accuracy of *CUIPs* generated by the proposed methods. The second data set is a much bigger data set harvested from the AOL search query log. This data set was used to test the improvement in personalized search for the two proposed methods, and their comparisons with other methods.
4. Our results show that personalized search using the *modSvdCUIP* is better than using the *tfUIP(term frequency UIP)*(Noll and Meinel, 2007) and

4.1 Aggregating tags from user search history

tfIdfUIP (term frequency Inverse Document Frequency UIP) (Xu, Bao, Fei, Su, and Yu, 2008), and exhibits modestly better performance than the *tfIdfCUIP* (Andriy, Jonathan, Bamshad, and Robin, 2008) and *svdCUIP*. Each cluster, in the cluster structure *CUIP*, identifies a topic, and the application of *CUIP* helps disambiguate the context of user query, which is particularly needed for vague queries.

4.1 Aggregating tags from user search history

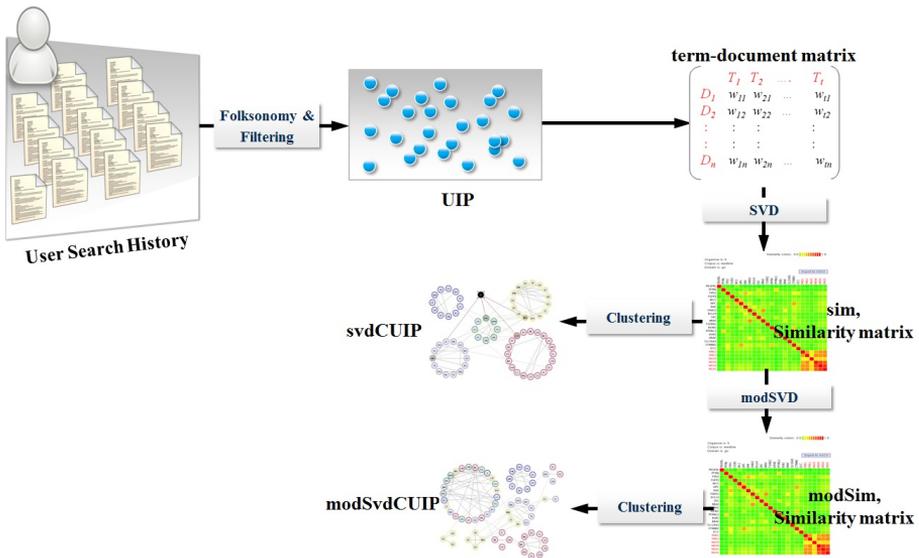


Figure 4.1: System Architecture of CUIP based Personalized Search

Figure 4.1 presents the overall architecture of CUIP based personalized

4.1 Aggregating tags from user search history

Table 4.1: Clicked Web documents and tags attached to the documents

URL	Tag
d_1	java, application
d_2	java
d_3	travel
d_4	iphone, game
d_5	iphone, application

search. When a user clicks on a Web document, it indicates the user interest in that document (Agichtein, Brill, and Dumais, 2006). A user search history provides a collection of the Web documents clicked by the user. Let's call the collection set U . For each Web document $u \in U$, its annotations (tags) are extracted from a social bookmarking service. The tags are stemmed during extraction. Let T be a set of stemmed tags extracted from the social bookmarking service. Note that it is not necessary for the user to have previously used these tags for annotation. The extracted tags were annotated to the documents by the users of the social bookmarking service. Let R be a binary relation between U and T . In order to express that a Web document $u \in U$ is in a relationship with a tag $t \in T$, we write $(t, u) \in I$, which can be read as "the tag t is a topic of the Web document u ". A user context in Table 4.2 is derivable from the relations between Web documents and the tags in Table 4.1. In Table 4.2, each row has a tag in its first column, followed by tag-values, each denoting the importance of the tag for the document clicked by the user. The higher the value, the more useful the tag is for describing the document. Each tag,

4.1 Aggregating tags from user search history

Table 4.2: A user context derivable from Table 4.1

	d_1	d_2	d_3	d_4	d_5
iphone	0	0	0	1	1
java	1	1	0	0	0
game	0	0	0	1	0
travel	0	0	1	0	0
application	1	0	0	0	1

t , annotated to a Web document, d_i , has a tag-value $w(t, d_i)$ representing the number of times d_i has been annotated with t . For example, $w(java, d) = 1$ means the tag *java* has been used to annotate the document d once. A tag weight, $w(t)$, is an aggregated value of t originating from the resource profiles (*RPs*) of multiple documents. It is very likely that the same tag may originate from multiple documents, each with a potentially different tag-value for the tag. We use the standard result set fusion technique, shown in Equation 4.1, to aggregate the tag weight, $w(t)$, from the Web document collection $|U|$.

$$w(t) = \sum_{i=1}^{|U|} w(t, d_i) \tag{4.1}$$

A *UIP* is constructed by collecting all the tags along with their tag weights. For example, the *UIP* for the user context in Table 4.2 would be [*java* : 2, *game* : 1, *application* : 2, *travel* : 1, *iPhone* : 2].

Similar to the well-known *term frequency * inverse document frequency* for documents in IR, the same can be modelled in constructing a *UIP*. The *tf*idf* multiplies the normalized tag frequency $\frac{td[i][j]}{|td[j]|}$ by the relative distinctness of

4.1 Aggregating tags from user search history

the tag $t[i]$ in the Web document corpus. The distinctness is measured by the log of the total number of Web documents, $|U|$, divided by the number of Web documents, $|\overrightarrow{td[i]}|$, to which the tag $t[i]$ was annotated to. We define the $tf * idf$ as follows.

$$td = \begin{matrix} & d_1 & d_2 & d_3 & d_4 & d_5 \\ \begin{matrix} iphone \\ java \\ game \\ travel \\ application \end{matrix} & \begin{pmatrix} 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 \end{pmatrix} \end{matrix} \quad (4.2)$$

$$tfIdf[i][j] = \frac{td[i][j]}{|\overrightarrow{td[j]}|} \log_2 \left(\frac{|U|}{|\overrightarrow{td[i]}|} \right) \quad (4.3)$$

Using Equation 4.3, the term-document matrix in Equation 4.2 is transformed to tfIdf Matrix, A , as follows.

$$A = \begin{matrix} & d_1 & d_2 & d_3 & d_4 & d_5 \\ \begin{matrix} iphone \\ java \\ game \\ travel \\ application \end{matrix} & \begin{pmatrix} 0 & 0 & 0 & 0.661 & 0.661 \\ 0.661 & 1.3219 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1.162 & 0 \\ 0 & 0 & 2.32 & 0 & 0 \\ 0.661 & 0 & 0 & 0 & 0.661 \end{pmatrix} \end{matrix}$$

4.2 Latent Semantics in UIP

Latent semantics connotes hidden relationships among terms that may exist, but are not explicitly visible. The latent semantics between terms can be discovered by observing the patterns between them such as co-occurrence. Extracting latent semantics between terms helps improve the usefulness of the *UIP*. Co-occurrence between tags can be classified into two types:

1. Two or more tags that annotate the same document: there exist first-order co-occurrences between the tags.
2. Two or more tags that do not annotate the same document; however, there is some hidden relationship between them because they may be related to similar topics: there exist second-order co-occurrences between the tags.

We propose a system that discovers semantically related tags and groups them together, even though they are not identical or do not annotate the same document. The approaches to establishing latent structures in a *UIP* are based on the assumption that the more similar tags are, the more closely related they are.

4.2.1 Computing the tag-tag Similarity matrix

Co-occurrence similarity derives similarity between two or more tags that annotate the same document. The degree of similarity is calculated using the

co-occurrence frequency, called first-order co-occurrence similarity. Another type of co-occurrence similarity is second-order co-occurrence similarity that derives similarity between two tags that do not annotate the same document, but both are related to at least one other tag that annotates the document. It is analogous to finding a friend of a friend and quantifies the degree of friendship relationship. A straightforward approach to measuring the similarity between two tags is to use the Jaccard coefficient between their tag vectors. An alternative approach is to employ matrix factorization on the tfIdf matrix.

We use two matrix-factorization-based methods to calculate the tag-tag similarity matrices. Latent Semantic Analysis (LSA) (Scott, Susan, George, Thomas, and Richard, 1990) uses a matrix factorization technique, Singular Value Decomposition (SVD), to create a new abstract representation of a document corpus in the latent squares sense. The SVD decomposes the tfIdf matrix into three matrices, $A = USV^T$: U , a tag by dimension matrix; S , a diagonal matrix of singular values; and V , a document by dimension matrix. The SVD translates the tag and document vectors into a space determined by the rank r of matrix A . The first r columns of matrix U and matrix V form an orthogonal basis for the tag by document matrix's tag space and document space, respectively

One advantage of the SVD is that it is possible to find a low-rank approximation of the original matrix that removes noise. When we select the k largest singular values from S and their corresponding singular vectors from U and V , we get the rank k approximation, $A_k = U_k S_k V_k^T$, where k is the dimension

reduction parameter. The left singular vectors provide a mapping from the tag space to a newly generated abstract space, while the right singular vectors provide a mapping from the document space to a newly generated space. To compute the tag-tag similarity matrix, we compute U_k , a low-rank approximation of U matrix. After the dimensionality reduction step, the term-term similarity matrix, Sim_k , is computed by using Equation 4.4.

$$Sim_k = U_k S_k (U_k S_k)^T = U_k S_k S_k^T U_k^T = U_k S_k^2 U_k^T \quad (4.4)$$

Dimensionality reduction reduces noise in the tag-tag similarity matrix, resulting in richer relationships between tags that reveals the hidden semantics present in the document corpus. The value of Sim_{ij} in Sim_k represents the similarity between tags t_i and t_j . The higher the value, the higher the relatedness is between the tags. In theory, the value of Sim_{ij} captures both orders of co-occurrence similarities between t_i and t_j across the corpus. That is, the value is based on the transitive relation between terms due to a chain of intermediate terms that link the terms t_i and t_j . Note that it is not necessary for t_i and t_j to belong to the same document, but there should be a chain of terms that link them. Two factors influence the magnitude of similarity value Sim_{ij} : 1) the number of intermediate tags, or the length of the chain that connects t_i and t_j ; and 2) the tag-weights of the intermediate tags. The example below shows the step-by-step procedures to obtain the similarity matrix, Sim_2 , by applying Equation 4.4 on the tfIdf matrix, A .

Note that there exists a disparity in the similarity values in Sim_2 . The reason is that the user context in Table 4.2 indicates that the tag "iphone" is co-located with the tags "game" and "application", and not with the tag "java". The SVD process has successfully captured the relationships "iphone" and "game", and "iphone" and "application", which is a first-order co-occurrence relationship. Also, it has successfully discovered the hidden relationship between "iphone" and "java", because of the intermediate tag "application" that co-occurs with "java" and "iphone". However, the magnitude of relationship is misleading: it suggests a stronger relationship between "java" and "iphone" (0.3517) compared to "iphone" and "application" (0.1235), and "iphone" and "game" (0.0481).

4.2 Latent Semantics in UIP

$$\begin{aligned}
 A &= \begin{matrix} & d_1 & d_2 & d_3 & d_4 & d_5 \\ \begin{matrix} iphone \\ java \\ game \\ travel \\ application \end{matrix} & \begin{pmatrix} 0 & 0 & 0 & 0.661 & 0.661 \\ 0.661 & 1.3219 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1.162 & 0 \\ 0 & 0 & 2.32 & 0 & 0 \\ 0.661 & 0 & 0 & 0 & 0.661 \end{pmatrix} \end{matrix} \\
 U &= \begin{pmatrix} 0.00 & -0.16 & -0.59 & 0.28 & -0.74 \\ 0.00 & -0.92 & 0.26 & -0.27 & -0.1 \\ 0.00 & -0.13 & -0.75 & -0.45 & 0.46 \\ 1.00 & 0.00 & 0.00 & -0.00 & 0.00 \\ 0.00 & -0.32 & -0.14 & 0.00 & 0.48 \end{pmatrix} \quad U_2 = \begin{pmatrix} 0.00 & -0.16 \\ 0.00 & -0.92 \\ 0.00 & -0.13 \\ 1.00 & 0.00 \\ 0.00 & -0.32 \end{pmatrix} \\
 S &= \begin{pmatrix} 2.32 & 0.00 & 0.00 & 0.00 & 0.00 \\ 0.00 & 1.53 & 0.00 & 0.00 & 0.00 \\ 0.00 & 0.00 & 1.4 & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.94 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.00 & 0.33 \end{pmatrix} \quad S_2 = \begin{pmatrix} 2.32 & 0.00 \\ 0.00 & 1.53 \\ 0.00 & 0.00 \\ 0.00 & 0.00 \\ 0.00 & 0.00 \end{pmatrix} \\
 Sim_2 &= U_2 S_2^2 U_2^T
 \end{aligned}$$

$$\begin{aligned}
 & \begin{matrix} & iphone & java & game & travel & application \\ \begin{matrix} iphone \\ java \\ game \\ travel \\ application \end{matrix} & \begin{pmatrix} 0.0621 & \mathbf{0.3517} & 0.0481 & 0.00 & \mathbf{0.1235} \\ 0.3517 & 1.9928 & 0.2726 & 0.00 & 0.6996 \\ 0.0481 & 0.2726 & 0.0373 & 0.00 & 0.0957 \\ 0.00 & 0.00 & 0.00 & 5.3914 & 0.00 \\ 0.1235 & 0.6996 & 0.0957 & 0.00 & 0.2456 \end{pmatrix} \end{matrix} \\
 &= \begin{matrix} & iphone & java & game & travel & application \\ \begin{matrix} iphone \\ java \\ game \\ travel \\ application \end{matrix} & \begin{pmatrix} 0.0621 & \mathbf{0.3517} & 0.0481 & 0.00 & \mathbf{0.1235} \\ 0.3517 & 1.9928 & 0.2726 & 0.00 & 0.6996 \\ 0.0481 & 0.2726 & 0.0373 & 0.00 & 0.0957 \\ 0.00 & 0.00 & 0.00 & 5.3914 & 0.00 \\ 0.1235 & 0.6996 & 0.0957 & 0.00 & 0.2456 \end{pmatrix} \end{matrix}
 \end{aligned}$$

could be beneficial for a clustering algorithm to generate accurate clusters.

$$Sim_3 = \begin{matrix} & \begin{matrix} iphone & java & game & travel & application \end{matrix} \\ \begin{matrix} iphone \\ java \\ game \\ travel \\ application \end{matrix} & \left(\begin{matrix} 0.3577 & \mathbf{0.0275} & \mathbf{0.4404} & 0.00 & \mathbf{0.1349} \\ 0.0275 & 1.0185 & -0.0491 & 0.00 & 0.3035 \\ 0.4404 & -0.0491 & 0.5488 & 0.00 & 0.1422 \\ 0.00 & 0.00 & 0.00 & 2.5903 & 0.00 \\ 0.1349 & 0.3035 & 0.1422 & 0.00 & 0.1354 \end{matrix} \right) \end{matrix}$$

However, even with $k = 3$, the magnitudes of relationship, expressed in similarity values, are rather low for second-order co-occurrence similarity ("iphone" and "java") compared to the first-order co-occurrence similarity ("iphone" and "game" or "iphone" and "application"). This seriously jeopardizes the effectiveness of the clustering algorithm to generate clearly separated clusters. In real scenarios, sparseness of a similarity matrix, Sim_k , could be as high as 90%, which seriously affects the ability of the SVD to correctly discover the second-order co-occurrences. We show in the experiment section the effect of sparseness of Sim matrices on clustering tendency and clustering accuracy.

The second-order co-occurrence similarity values are too small to be detected by clustering algorithms. The experiment results show that the numbers of values in the term-term similarity matrix, greater than 0.5, is small, nullifying the usefulness of SVD to discover 2^{nd} order co-occurrence between terms.

Higher values of $modSim_{ij}$ signify a greater overlap between the two vectors across n dimensions. Thus, it aids in demarcating clusters boundaries, resulting in fine clusters, and also helps induce sense from contextual similarity.

4.2.2 Tag Clustering to generate *svdCUIP* and *modSvdCUIP*

(Scott, Susan, George, Thomas, and Richard, 1990) urged the necessity of clustering in Information Retrieval (IR) tasks. The authors state that IR systems treat each term as independent from others. Treating a term independently may lose the latent contextual information that can make substantial difference in information retrieval tasks. This has motivated us to use clustering in our work.

Term Clustering algorithms generally consist of two phases. The first phase requires computing a term-term similarity matrix, and the second phase uses the matrix to generate clusters of coherent terms. Two major types of clustering algorithms are available: partitioning and hierarchical. The partitioning clustering generates topic clusters, whereas the hierarchical clustering generates cluster hierarchies. Topic clusters are created by grouping similar and closely related terms together into a unified topic. In a cluster hierarchy, terms are placed in the leaves at the bottom of the hierarchy with more specialized topics immediately above them, and so on. Hierarchies are very large and complex in nature. We want hierarchies but not too specific terms. We are, on the other hand, interested in crisp clusters. Therefore, we adapted a hybrid approach that generates a hierarchy, which is further dissected to generate crisp

term clusters. We used the Hierarchical Agglomerative Clustering Algorithm (HAC)(Gower and Ross, 1969) because it fits best when the number of clusters is unknown beforehand. We use distinctness parameter, d , to cut the single hierarchy of clusters to obtain a number of clusters. For instance, Table 4.3 shows the clusters, in the cluster structures *svdCUIP* and *modSvdCUIP*, obtained by applying HAC on *sim₃* and *modSim₃* matrices. The *svdCUIP* has four clusters, and it fails to identify that "iphone" and "game" should belong to the same cluster, whereas the *modSvdCUIP* identifies all the term clusters accurately. It is very important to choose the right value of d to generate appropriate term clusters matching the user's perspective, thus achieving a high clustering accuracy. Figure 4.2 shows a dendrogram output when the similarity matrix *modSim* is input to the HAC. With $d \geq 1.4$, one cluster is created, a hierarchy of all the terms; with $d = 0.4$, there are three clusters; and, with $d < 0.3$, there is a flat list of terms.

At the outset, HAC treats each term as a singleton cluster and then successively merges pairs of clusters until all the clusters have been merged into a single cluster that contains all the terms. Cluster proximity is used to merge clusters. There are three well known proximity measures: single linkage, complete linkage, and average linkage. The single linkage proximity measure is the distance between the closest two points that are in two different clusters, i.e., the maximum similarity between two terms. On the contrary, the complete linkage takes the distance between the farthest two points in two different clusters as the cluster proximity. The average linkage defines cluster proximity as

Table 4.3: Clusters obtained by applying HAC on similarity matrices Sim_3 and $modSim_3$ for $k=3$ and $d=0.35$

Method	Cluster Structure
<i>svdCUIP</i>	[[iphone], [java, application],[game],[travel]]
<i>modSvdCUIP</i>	[[java, application],[iphone,game],[travel]]

the average pairwise proximity, an average length of edges of all the terms from two different clusters. We carried out experiments using the three proximity measures, but this research reports on only the average linkage in the experiment section because it worked better than the others. The explanation in the previous two sub-sections has identified the importance of dimensionality reduction parameter k and distinctness parameter d to generate right number of clusters of good quality. The experiment section shows how to determine the right values of k and d , to generate crisp clusters, without compromising clustering accuracy.

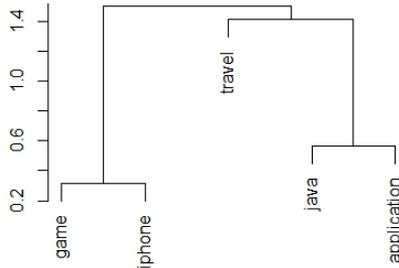


Figure 4.2: Dendrogram visualization for similarity matrix $modSim$

A *CUIP* that results from the application of HAC on a *Sim* matrix obtained

by applying the SVD on a *tfidf* matrix is called SVD based CUIP (*SvdCUIP*). And, a *CUIP* that results from the application of HAC on a *modSim* matrix obtained by calculating the cosine similarity of every pair of tag vectors in the similarity matrix, *Sim*, is called modSVD based CUIP (*modSvdCUIP*).

We also generate a *tfidfCUIP* for each user, an adaptation of (Andriy, Jonathan, Bamshad, and Robin, 2008) approach. A term-term similarity matrix is generated by computing the cosine similarity between tag vectors in the *tfidf* matrix, which is fed to HAC to generate the *tfidfCUIP*. The *tfidfCUIP* is a local cluster structure unlike the (Andriy, Jonathan, Bamshad, and Robin, 2008) approach where the terms in the *UIP* are mapped to a global cluster structure to construct a *CUIP*.

4.3 Personalized Search

This section explains how to use a CUIP for personalized search. The classic SEs compute the relevance between a query and a document using the similarity between the terms that match. They are "One-size-fits-all" in that the search results are the same irrespective of the user. However, a document relevant to a user might not be relevant to another user, though, they both have issued the same query. Thus, the user query as well as its context should be mapped to the term space of the document contents. A query conflated with the contextual terms is called expanded query.

The *CUIP* helps disambiguate a user query by suggesting a matching cluster.

The terms in Web documents and the expanded query are represented as vectors in the space. By using the Vector Space Model (VSM) (Salton, Wong, and Yang, 1975), we compute the similarity between the term spaces of the documents and that of the expanded query to compute the rank of the documents. Let $d = t_1^d, t_2^d, \dots, t_n^d$ be the term vector for a document, where n is the dimension of the term space. Let $qe = t_1, t_2, \dots, t_n$ be the expanded query. The similarity between a document d and a query qe is calculated using Equation 4.6.

$$sim(d, qe) = \frac{d^T \cdot qe}{|d||qe|} \quad (4.6)$$

Given a user query, two steps are executed in the following order: first, find a matching cluster g_m in the user *CUIP* to the query; second, the query and the tags in the matching cluster are fed to the underlying search engine to generate a set of documents that are ranked using equation 4.6.

In this research, we use a class-based Language Modeling (LM) to determine the most closest cluster, for a given query, from the user's cluster structure. This involves computing the similarity between each cluster and query, and choosing the cluster that has the maximum similarity, refer equation 4.3).

$$CUIP = \{g | g = \{t_1, t_2, \dots, t_n\}\}$$

$$P(q, CUIP) = \underset{g \in CUIP}{argmax} \prod (q|g)$$

$$P(q, g) = P(q|t_1, t_2, \dots, t_n)$$

$$P(q|t_1, t_2, \dots, t_n) = \prod_{i=1..n} P(q|t_i)$$

where

$$P(q|t_i) = \frac{\text{count}(q, t_i)}{\text{count}(t_i)} \quad (4.7)$$

4.4 Experimental Evaluation

4.4.1 Data Set and Experiment Methodology

To examine the effectiveness of the proposed methods, we conducted a series of experiments on two different data sets. First, to evaluate the clustering tendency and clustering accuracy of the *CUIP*, we recruited 12 users whose search histories were harvested to construct the first data set, referred as Custom Data Set. Second, to evaluate the quality of personalized search using the proposed methods, we constructed another data set from the AOL search query log¹. For both data sets, the URL-tag annotations were harvested from the Delicious Server using the Delicious API².

4.4.1.1 Custom Data Set and Evaluation Metrics

This data set consists of data from 12 users, mostly master's students, who have considerable experience using search engines. Each user's log of search history for a period of 3 months or 13 weeks was harvested as an RSS feed

¹<http://www.gregsadetsky.com/aol-data/>

²<http://www.delicious.com>

4.4 Experimental Evaluation

from the individual’s Google Search History¹. The RSS feed consists of the following meta data: title of the query input by the user; title of the Web document clicked by the user; the address of the Web document clicked by the user; and, the dates and times at which the queries were submitted. The data set contains 2921 queries, and 6477 clicked Web documents. Of the documents, only 3617 (approximately 55%) were found to be annotated on Delicious.

In clustering, measuring its accuracy and correctness in any certainty is best left to the user’s judgement. Therefore, to establish the ground truth, we asked each user to group related terms extracted from the tag annotations of the Web documents clicked by the user. Each user was asked to manually group related terms together; they were instructed to group terms based on their own understanding rather than the general understanding. The grouping generated manually by the user is called user cluster structure. Generating ground truth manually for evaluation is a normal procedure used in many research works (Bing, 2006, Christopher, Shlomo, and Andrew, 2012, Dom, 2002, Hassan, 2006, Pérez, Zubiaga, Fresno, and Martínez, 2012). Since this process is subjective, we take the average of the scores from all the users as the final score. The whole process was a very labor intensive and time consuming task, which was the primary reason why we opted to experiment with a small set of users.

For each user, two sets of *CUIPs* are generated: one set consists of *svdCUIPs*, and the other of *modSvdCUIPs*. These *CUIPs* are called system generated

¹<http://www.google.com/searchhistory>

4.4 Experimental Evaluation

cluster structures. In each set, a *CUIP* is generated for each combination of dimension reduction parameter k and distinctness parameter d . To construct a *svdCUIP* and a *modSvdCUIP*, the similarity matrices sim_k and $modSim_k$ are generated, respectively. The value for k is initialized to 10, and it increases in an increment of 10 until it reaches 110. This creates 11 sim_k and 11 $modSim_k$ similarity matrices. Similarly, the distinctness parameter d is initialized to 0.03, and it increases in an increment of 0.02 until 0.13, after which it increases in an increment of 0.1 until 0.93 (a total of 14 values). For each user, 154 *svdCUIPs* and an equal number of *modSvdCUIPs* were created. Let the user generated cluster be $C = \{c_1, c_2, \dots, c_n\}$, and the system generated cluster be $D = \{d_1, d_2, \dots, d_m\}$. We chose the Silhouette Coefficient (Rousseeuw, 1987) evaluation metric (unsupervised evaluation) to judge the cluster tendency, and the Fscore (supervised evaluation) evaluation metric to compare the clustering accuracy. The Silhouette Coefficient is a popular method that combines cohesion and separation. Equation 4.8 computes the Silhouette Coefficient for each tag t_i in the system cluster structure.

$$s(i) = \frac{(b_i - a_i)}{\max(a_i, b_i)} \quad (4.8)$$

where b_i is the minimum of all the average distances between term t_i and all the terms in other clusters that do not contain t_i (separation); and, a_i is the average distance between term t_i and all other terms in the same cluster (cohesion). Equation 4.9 computes the average Silhouette Coefficient, \bar{s} , which

4.4 Experimental Evaluation

is the average of the Silhouette Coefficients for all the terms (N) in the cluster structure.

$$\bar{s} = \frac{1}{N} \sum_{i=1}^n s(i) \quad (4.9)$$

An average Silhouette Coefficient is a very useful overall quality measure to measure the clustering tendency of a cluster structure. (Kaufman and Rousseeuw, 1990) provided an interpretation of the average Silhouette Coefficient, \bar{s} , as a measure of evidence in support of a cluster structure: the value of the average Silhouette Coefficient between]0.7, 1.0] suggests strong evidence; between]0.5, 0.7] reasonable evidence; between]0.25, 0.5] weak evidence; and between [-1, 0.25] no evidence.

We also compare the clustering accuracy of the system generated cluster structure with the user generated cluster structure. Fscore(Bing, 2006) measures the extent to which a system generated cluster contains only tags of a particular user generated cluster and all objects of that user generated cluster. Equation 4.10 computes an Fscore by combining precision and recall. Precision, p_i , is the proportion of the tags of user generated cluster c_j in the system generated cluster d_i ; Recall, r_i , is the fraction of matching tags in the system generated cluster d_i that match the tags in the user generated cluster c_j .

$$Fscore_i = \frac{2 * p_i * r_i}{p_i + r_i} \quad (4.10)$$

4.4.1.2 AOL Query Data Set and Evaluation Metrics

The AOL search query log has 20 million Web queries collected from 650,000 users. Each row in the data set contains five attributes: 1) AnonID, an anonymous user id; 2) Query, the query issued by the user; 3) Query Time, the time at which the query was submitted to the AOL search engine; 4) Item Rank, the rank of the Web document clicked by the user; and 5) ClickURL, the address of Web document clicked by the user. We created a dataset of 2000 users, a subset of the total data set. This dataset contains 1,244,714 Web documents, out of which 829,285 documents (approximately 66%) were found to be annotated on the Delicious server. The documents have 212,011 tags annotated to them. Our experiment methodology is geared towards measuring the effectiveness of the proposed personalized search methods and evaluating the improvement they offer in comparison to other methods. Figure 4.3 illustrates the overall evaluation methodology.

4.4.1.3 Experiment set up to estimate the value of k and d

The complete data set is split into two equal parts: the first part is called as the training, or development, data set; and the second part is called as the evaluation data set. The training data set is used to estimate the value of parameters k and d for *svdCUIP* and *modSvdCUIP*, which are directly used in the evaluation dataset to compare the performance of the proposed approaches with the other personalized search approaches. The evaluation data set helps

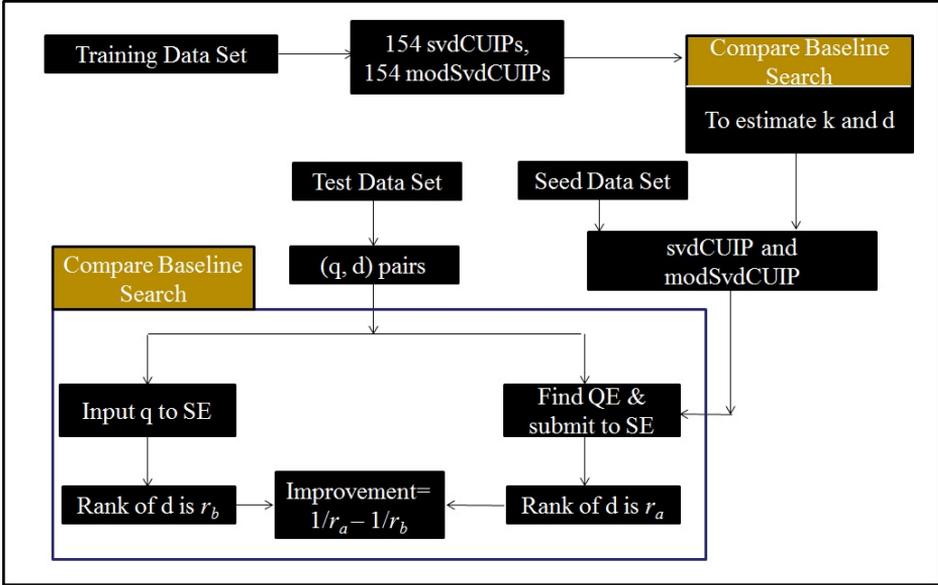


Figure 4.3: Automatic Evaluation Methodology

guard against both under fitting and over fitting.

From the training data set, we construct *UIPs* and *CUIPs*, and pairs of query and associated Web document (referred as target Web document) are extracted from the user search history. For each pair, the query is submitted to the base search engine to calculate the rank of the target Web document, called r_b . Next, the query is expanded with the tags in the matching cluster from the *CUIP*. The expanded query is submitted to the search engine to calculate the new rank of the target Web document, called r_a . The difference in the inverse ranks of the personalized search method and the baseline method is the improvement (Ellen,

1999) of the personalized search method, calculated using equation 4.11.

$$improvement = \frac{1}{r_a} - \frac{1}{r_b} \quad (4.11)$$

The values of k and d , for which the improvement of the proposed methods over baseline search is maximum, are used directly for the further stage of evaluation.

4.4.1.4 Experiment set up to compare the proposed approaches with other approaches

The following steps execute on the evaluation data set:

1. **Indexing:** The contents of each document in the dataset is indexed using Lucene API¹. Lucene is our base search engine, and search using it is referred to in this chapter as baseline search method.
2. **User Profile:** The search history of each user is divided into two parts: the first part, which makes 90% of the entire history, is used for building *UIPs* and *CUIPs*; and the second part, the remaining 10%, is used for generating pairs of queries and URLs, called test collection, to automatically evaluate our methods.
3. **Evaluation:** For each document in the second part, we create a pair that consists of the document itself and the query associated with it. Each

¹lucene.apache.org/core/

4.4 Experimental Evaluation

pair constitutes a test case against which the tasks (a), (b), (c), and (d) below are executed. A test case designates a query and its target Web document.

- (a) For each query and Web document combination in a test case, submit the query to the base search engine to obtain a ranked list of search results. Let the rank of the target Web document in the search result set be r_b . This is the rank of the target document produced by the baseline search method.
- (b) For both *tfUIP* and *tfIdfUIP*, the Web documents in the search result set are re-ranked by calculating the similarity between the *RP* of the Web documents and tags in the *UIP* using equations 2.1 and 2.2, respectively. Let the new ranks of the target document in the re-ranked search result set designated as r_n and r_x for *tfUIP* and *tfIdfUIP*, respectively. Equation 4.11 computes the improvement as the difference between the inverse ranks of the personalized search method and the baseline method.
- (c) Search results are not re-ranked for the *svdCUIP*, *modSvdCUIP*, and *tfIdfCUIP* methods, rather, the query is expanded with the tags in the matching cluster from the *CUIP*. The expanded query is submitted to the search engine to determine a new rank of the target document. The search engine generated the ranking of documents by calculating the similarity between the expanded query and the

document contents using the equation 4.6. The difference in the inverse ranks determined for the personalized search method and the baseline method is the improvement of the personalized search method.

4.4.2 Experiment Results

Sections 4.4.2.1, 4.4.2.3, and 4.4.2.3 determine, for both *svdCUIP* and *modSvdCUIP*, the value(s) of dimensionality reduction parameter k and distinctness parameter d that show(s) strong, or at least reasonable, clustering tendency and clustering accuracy. Section 4.4.2.4 presents an exemplary *modSvdCUIP*. The sections 4.4.2.5 and 4.4.2.6 determine, for both *svdCUIP* and *modSvdCUIP*, the value(s) of dimensionality reduction parameter k and distinctness parameter d using the Improvement as an evaluation metric. And, sections 4.4.2.8 and 4.4.2.9 compare the proposed methods with the other methods using the evaluation metric Improvement.

4.4.2.1 Clustering Tendency

Assessing the presence of clusters in a data set is an important step in cluster analysis. The plot in Figure 4.4 helps visualize clustering tendency in system generated clusters, if any, and also approximates the correct number of clusters in the cluster structure.

It is clear that the cluster structure *modSvdCUIP* has stronger evidence of cluster tendency, whereas the *svdCUIP* shows reasonable or weak evidence of

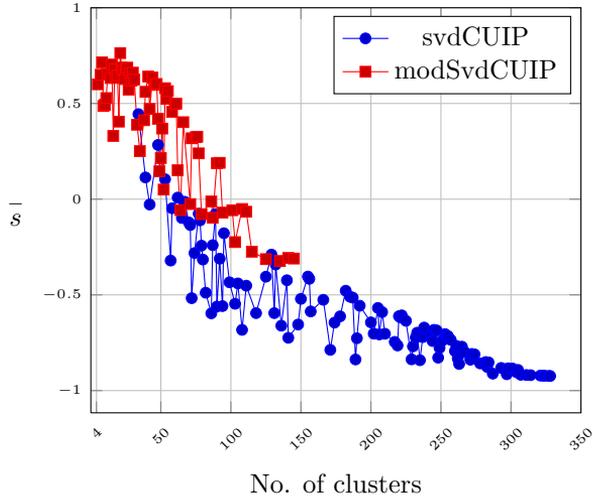


Figure 4.4: Number of Clusters vs. average Silhouette Coefficient plot for *svdCUIP* and *modSvdCUIP*

clustering tendency. We observed that the clustering tendency in a *CUIP* was affected by the ratio of number of zero values to the number of positive values in the tag-tag similarity matrix; the lower the better. The average ratio for the tag-tag similarity matrix *modSim* is 0.9, and 1.68 for the tag-tag similarity matrix *sim*. The maximum and minimum ratios for the *modSim* are 3.2 and 0.6, respectively, and for the *sim*, 6.2 and 1.0, respectively. This evidence explains why the cluster structure, *svdCUIP*, exhibits weak cluster tendency.

Figure 4.4 also indicates that the average Silhouette Coefficient (\bar{s}) decreases as the number of clusters exceeds over 50, which suggests that the best cluster structure was obtained when the number of clusters was around 50. This was

acceptable because the average number of tags in a *UIP* was 594, which could possibly result in 50-70 clusters. However, what is surprising is that, even with less than 10 clusters in the *modSvdCUIP*, the plot shows strong clustering tendency. To try to find the natural number of clusters in a cluster structure, one should look for a knee, a peak, or dip in the plot (Tan, Steinbach, and Kumar, 2005). The plot for the *modSvdCUIP* shows a rise followed by a dip and a peak occurring around when the number of clusters falls between 40 and 60. For the *svdCUIP*, the plot clearly shows a peak when the number of clusters reaches 50.

4.4.2.2 Determining the value for dimension parameter, k , for the Custom Data Set

Figures 4.5 and 4.6 present 3-dimensional plots that show how the average Silhouette Coefficient changes in response to the changes of k and d . The figures help determine the values of k and d for each method. The *svdCUIP* in Figure 4.5 exhibits a clear pattern: for low values of k regardless of d , there is no evidence of clustering tendency; however, for high values of k , between 90 and 100 and low values of d , there is a reasonable evidence of clustering tendency. The weak clustering tendency of the *svdCUIP* is due to the fact that the magnitude of relationship between tags is low. This jeopardizes the ability of clustering algorithms to discern cluster boundaries.

The average Silhouette Coefficient vs. k and d plot in Figure 4.6 for the cluster structure *modSvdUIP* also exhibits a distinct pattern: unlike the *svdCUIP*,

4.4 Experimental Evaluation

the plot for the *modSvdCUIP* shows a strong evidence of clustering tendency for values of $k = 30$ and 40 and middle values of d . It ascertains the fact that increasing the value of d decreases clustering tendency. The *modSvd-CUIP* exhibits a strong clustering tendency because the *modSim* overcomes the limitation of the *Sim* by capturing the information present in second order co-occurrence. Moreover, the information in the *modSim* matrix is less sparse and more robust than the *Sim* matrix.

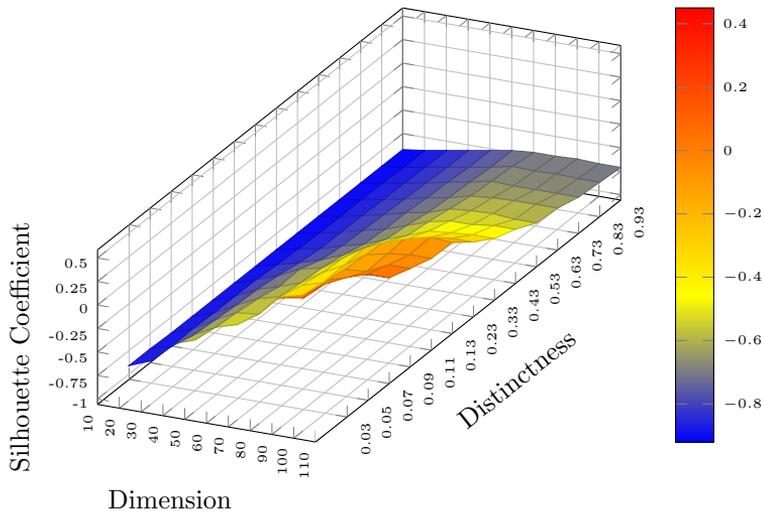


Figure 4.5: A comparison of different value combinations of k and d Vs. average Silhouette Coefficient for *svdCUIP* average linkage

4.4.2.3 Determining the value of distinctness parameter, d , for the Custom data set

The experiment, in this section, focuses on determining the appropriate value of d for the highest accuracy cluster structure. Fscore is used as an evaluation metric to measure and compare the accuracy of the system generated cluster structure with the user generated cluster structure. Figure 4.7 shows the accuracy obtained by each method, and demonstrates that the *modSvdCUIP* has better clustering accuracy than the *svdCUIP*.

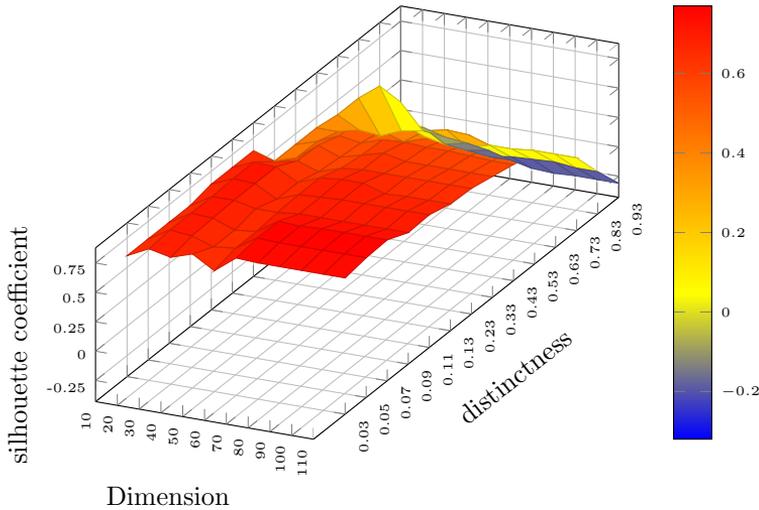


Figure 4.6: A comparison of different value combinations of k and d vs average Silhouette Coefficient for *modSvdCUIP* average linkage

The average clustering accuracy for the *modSvdCUIP* and *svdCUIP* is 0.58

4.4 Experimental Evaluation

and 0.16, respectively; there is a 244% increase in average clustering accuracy. This indicates that the *modSvdCUIP* produced by the *modSvd* is more accurate than the *svdCUIP* produced by the *Svd*. With the *modSvd*, the dimension reduction parameter $k=30$ has higher clustering accuracy than $k=40$. Also, the difference in clustering accuracy between $k=30$ and $k=40$ is marginal. Moreover, both of the curves follow the same pattern, signifying that the clustering accuracies of the *modSvdCUIP* for $k=30$ and $k=40$ are nearly identical with a slightly better performance at $k=30$. The highest clustering accuracy for the *modSvdCUIP* is 0.75, obtained with $k=30$ and $d=0.07$.

Another identical accuracy was exhibited when $k=90$ and $k=100$ in the *Svd*. A careful observation, however, reveals that the *svdCUIP* for $k=100$ shows a marginal improvement over $k=90$, with $d=0.03$ and $d=0.05$. This suggests that either value of the dimension reduction parameter can be used for constructing the *svdCUIP*. The highest clustering accuracy for the *svdCUIP* is 0.55, with $k=100$ and $d=0.03$.

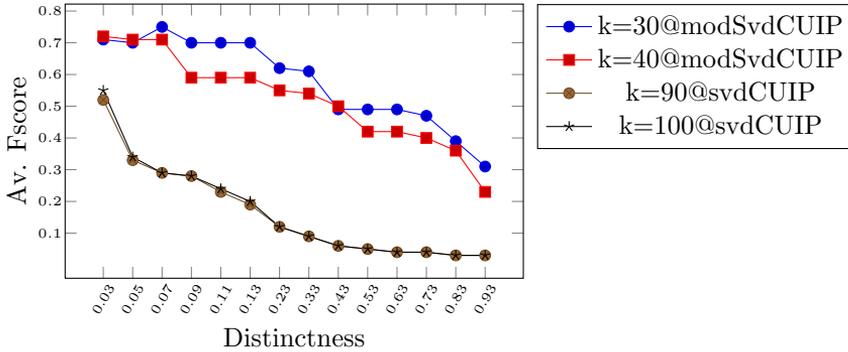


Figure 4.7: A comparison of different value combinations k and d vs *AverageFscores* for the *modSvdCUIP* (when $k=30,40$) and the *svdCUIP* (when $k=90,100$) for average linkage.

These results suggest that the accuracy of the *modSvdCUIP* produced by the *modSvd* is superior to the cluster structure *svdCUIP* produced by the *SVD*.

4.4.2.4 CUIP visualization

We developed our own implementation of Hierarchical Agglomerative Clustering (HAC) in Java. Table 4.4 shows the snapshot of the *modSvdCUIP*, the output of HAC for $d=0.53$, for one of the users. For interested readers, a complete *modSvdCUIP*, *svdCUIP*, and *tfIdfCUIP* is provided in the .3.

The quality of clusters hinges on the level of term coherency, each cluster representing a distinct topic area. Table 4.4 shows a high level of term coherency in clusters, each of which shows user interests such as finance, religion, porn, law, automotive, and entertainment. Moreover, the terms in each cluster are

Table 4.4: Example of cluster structure

Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5	Cluster 6
bank, banking, finance, business, supplier	religion, culture, judaism, jewish, israel	amateur, sex, adult, toys, girls, porn, voyeur	government, patent, trademark, law, legal	auto, automotive, parts, electronics, car	video, movies, film, soccer, game

contextually related, which aids to disambiguate context, synonym terms, and polysemous terms. For instance, Cluster 1 captures the notion of the user's interests in finance, and disambiguates the context of the polysemous term "bank", which in Cluster 1 refers to a financial institution, not to other meanings such as bank as in a river bank.

Cluster 2 indicates that the user is interested in Judaism religion. Synonym terms are clustered together such as "Jewish" and "Judaism" in Cluster 2, "auto" and "automotive" in Cluster 5, "movies" and "film" in Cluster 6. Cluster 5 can be interpreted as that the user is interested in the automotive, in particular cars. She/he might also be interested in the electronic parts of the car. Cluster 6 represents the user's entertainment options; the user prefers to watch movies or soccer games. The term video is rightly disambiguated by being associated with the term "movie".

These results show clear evidence of emergence of topics and contexts that would otherwise be latent in a *UIP*. A *CUIP* is an important source of information that can be effectively used for query suggestion, query classification, Web page recommendation, personalized search, or Web search result ranking.

4.4.2.5 Determining the value of the dimension reduction parameter k for the AOL data set

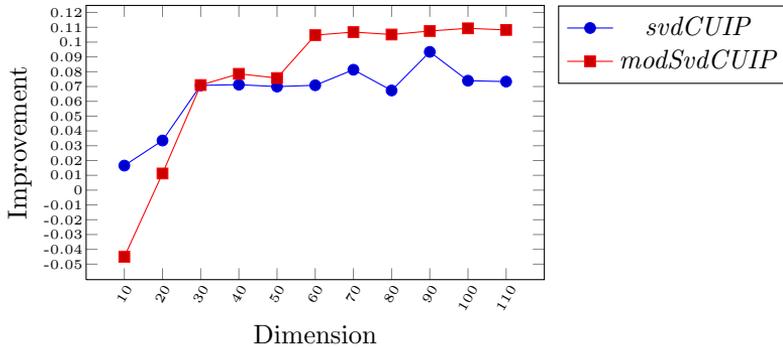


Figure 4.8: Estimating the values of dimension parameter for $svdCUIP$ and $modSvdCUIP$ using the Improvement as an evaluation metric

Since the personalization algorithm relies on the user $CUIP$ to personalize search results, the selection of a proper dimension value is integral to the success of the personalization algorithm. The goal of tuning the dimension parameter is to discover the second order co-occurrence similarity between tags. Figure 4.8 plots the improvement of proposed methods in reference to the baseline search when the value of k changes from 10 to 110 in an increment of 10. It indicates that the $modSvdCUIP$ based personalized search shows greater improvement than the $svdCUIP$ based personalized search. In this experiment, the most improvement was obtained when the value of k for the $svdCUIP$ and $modSvdCUIP$ was 90 and 100, respectively. Note that in a reduced space, the

4.4 Experimental Evaluation

performance of the *modSvdCUIP* based personalized search degraded below 0; this means that it performed worse than the baseline search. However, when k was set to 50 and above, it showed improved performance.

These results show that both methods benefited from the dimensional reduction step. In the following experiments, the value of k for the *svdCUIP* and *modSvdCUIP* was set to 90 and 100, respectively.

4.4.2.6 Determining the value of distinctness parameter, d , for the AOL data set

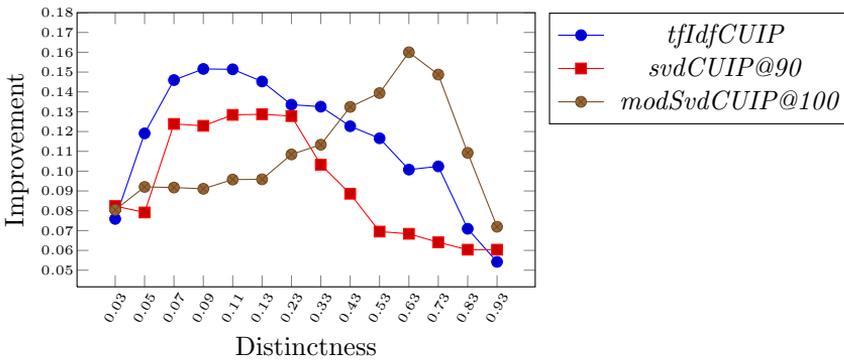


Figure 4.9: Estimating the values of distinctness parameter for *tfIdfCUIP*, *svdCUIP@90*, *modSvdCUIP@100* using Improvement as an evaluation metric.

The distinctness parameter d , controls how distinct or well separated the clusters are. As the value decreases, we get closer to a single cluster or a few large clusters; hence, grouping unrelated terms together or spanning multiple topic

4.4 Experimental Evaluation

areas. On the contrary, as the value increases, we end up with lots of clusters of a single term or lots of small-sized clusters, thus rendering the information in the clusters inadequate to represent topics. The parabolic graph in Figure 4.9 supports this idea. Note that there is no dimension reduction applied to the *tfIdfCUIP* method.

Figure 4.9 also shows that the *modSvdCUIP* based personalized search outperformed the *tfIdfCUIP* and *svdCUIP*. The maximum Improvement was obtained when d was set to 0.09, 0.13, and 0.63 for the *tfIdfCUIP*, *svdCUIP*, and *modSvdCUIP*, respectively. Performance of each CUIP is related to the number of clusters and the size of each cluster. The number of clusters for the *tfIdfCUIP* with $d=0.09$ is 54, 89 for the *svdCUIP@90* with $d=0.13$, and 76 for the *modSvdCUIP@100* with $d=0.63$. Also, the average number of tags in each cluster, average cluster size, for the *tfIdfCUIP* with $d=0.09$ is 6, 3 for the *svdCUIP@90* with $d=0.13$, and 4 for the *modSvdCUIP@100* with $d=0.63$. In short, having too many clusters, with only a few tags in each cluster, does not help disambiguate topics; this justifies why the *tfIdfCUIP* and the *modSvdCUIP* performed better than the *svdCUIP*.

In the following experiments that will execute on the evaluation data set, the value of d was set to 0.09 for the *tfIdfCUIP*, $k=90$ and $d=0.13$ for the *svdCUIP*, and $k=100$, $d=0.63$ for the *modSvdCUIP*.

4.4.2.7 Time to generate *svdCUIP* and *modSvdCUIP*

The aim of the experiment is to learn how much average time it takes to generate a *CUIP*. The results are plotted in Figure 4.10.

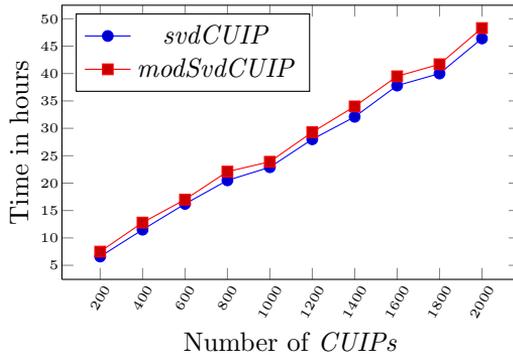


Figure 4.10: Average time to generate *svdCUIP* and *modSvdCUIP*

It shows that time to generate *CUIPs* is linear in nature. It took 46.4 and 48.3 hours to generate 2000 *svdCUIPs* and *modSvdCUIPs*, respectively, one for each individual user. In other words, a *svdCUIP* for a user can be generated in 83.52 sec, whereas a *modSvdCUIP* for a user can be generated in 86.94 sec. The difference is not huge. Note that, the generation of a *CUIP* is a background process so effectively it doesn't hurt the on-line execution time. Moreover, the time to generate a *CUIP* can be exponentially scaled down by using Mahout API that executes HAC on a hadoop cluster. We have already taken this viewpoint into consideration, therefore, since beginning all data at various stages is stored in csv file format.

4.4.2.8 Comparison of the *svdCUIP*, *modSvdCUIP*, and *tfIdfCUIP* for different classes of queries

The purpose of using the *modSvdCUIP* for personalized search is to identify the query context that we supposed the *tfIdfCUIP* would not be able to provide. However, the results presented in the previous sections indicate that the personalized search based on the *modSvdCUIP* and *tfIdfCUIP* delivered comparable effectiveness in improving the ranks of target Web documents. To further look into the effect that clusters have on personalized search, we analyzed the test collection, and found that self-evident queries didn't require disambiguation, and some vague queries received benefit when contextual tags were conflated with them. We identified 40 vague queries and 50 self-evident queries (refer to Appendix .1). Appendix .2 shows some examples of expanded queries and how query disambiguation is useful to personalized search.

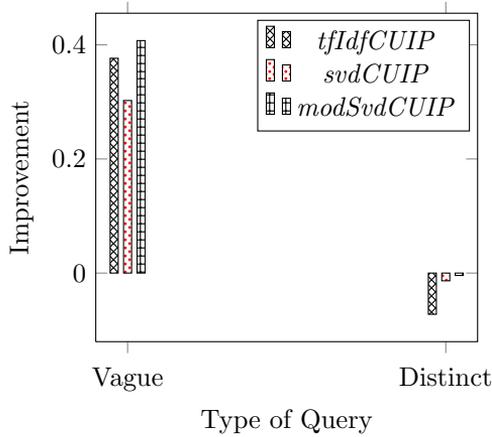


Figure 4.11: Comparing the Percentage Increase of the *tfIdfCUIP*, *svdCUIP*, *modSvdCUIP* for two classes of queries: vague and self-evident.

Figure 4.11 shows that the *modSvdCUIP* performed significantly better than both methods for the vague queries. And any modification of the self-evident queries by query expansion degraded the performance of the *CUIP* based personalized search methods. The *tfIdfCUIP* had the worst negative effect when used for disambiguating self-evident queries because the average cluster size is larger compared to other methods, thus degrading the ranks of the target Web documents.

4.4.2.9 Comparing all five methods - Improvement

This experiment aims to compare our proposed two methods with the others:
 1) tf based personalized search, *tfUIP*; 2) tfIdf based personalized search, *tfId-*

fUIP; and 3) *tfIdfCUIP* based personalized search.

As shown in Figure 4.12, the worst performer is the *tfIdfUIP*, similar to as reported by (David, Iván, and Joemon, 2010); results of both this study and (David, Iván, and Joemon, 2010) contradict those of (Xu, Bao, Fei, Su, and Yu, 2008) that the *tfIdfUIP* performed better than the *tfUIP*. A possible reason for the contradiction between ours and (Xu, Bao, Fei, Su, and Yu, 2008) approach is the total size of the result set; (Xu, Bao, Fei, Su, and Yu, 2008) re-ranked the top 100 Web documents, whereas our methods calculated the re-rank of the target URL in the top 600 documents. We suppose that the *tfUIP* showed better improvement than the *tfIdfUIP* because of the exclusion of two factors from the similarity score computation: document length and user profile length normalization factors. The user profile length normalization factor is dominant in the *tfIdfUIP*, and this penalizes the re-ranking score extensively.

The maximum improvement of the *modSvdCUIP* was 0.176766, whereas for the *svdCUIP* and the *tfIdfCUIP* was 0.132146 and 0.155571, respectively.

We performed significance test to determine if the difference between observed values from each approach are significant when compared with the baseline search. We used paired sample t-test and compared the average MRR values. Table 4.5 shows that the differences between the values from the *tfIdfUIP*, *tfUIP*, *tfIdfCUIP*, *svdCUIP*, and *modSvdCUIP* are significantly better than the baseline search. The MRR values were confirmed to be significantly different using the paired t-test with 95% confidence interval: *tfIdfUIP*(p-value=1.87E-09), *tfUIP* (p-value=1.67E-10), *tfIdfCUIP*(p-value=4.1E-

4.4 Experimental Evaluation

11), *svdCUIP*(p-value=4.2E-10), *modSvdCUIP*(p-value=2.31E-12). Thus, we can confidently conclude that the improvement of our proposed approaches is better than the baseline search.

	<i>tfIdfUIP</i>	<i>tfUIP</i>	<i>tfIdfCUIP</i>	<i>svdCUIP</i>	<i>modSvdCUIP</i>
MRR	0.3434	0.3625	0.4118	0.3946	0.4243

Table 4.5: Comparing the MRRs of *tfIdfUIP*, *tfUIP*, *tfIdfCUIP*, *svdCUIP*, and *modSvdCUIP*

4.4.3 Discussion

The strength of personalized search based on a *modSvdCUIP* lies in the discovery of second order similarity between tags, which is credited to the *modSim* tag-tag similarity matrix. The modSvd method generates a *modSvdCUIP* by applying HAC algorithm on the *modSim* matrix, which aids in discriminating tag sense by clustering semantically related tags together regardless whether they were originally collocated or not. Each cluster is assumed to correspond to a topic or to a sense of ambiguous tags. The poor result of personalized search based on *svdCUIP* is because it generated many small-size clusters resulting in inadequate disambiguation of user queries.

The best performance of *modSvdUIP* for the custom data set was observed when the dimension parameter k was set to 30. The average document space of the custom data set is 300, which is the average number of Web documents clicked by the users, and a reduced dimension space of 30 results in better

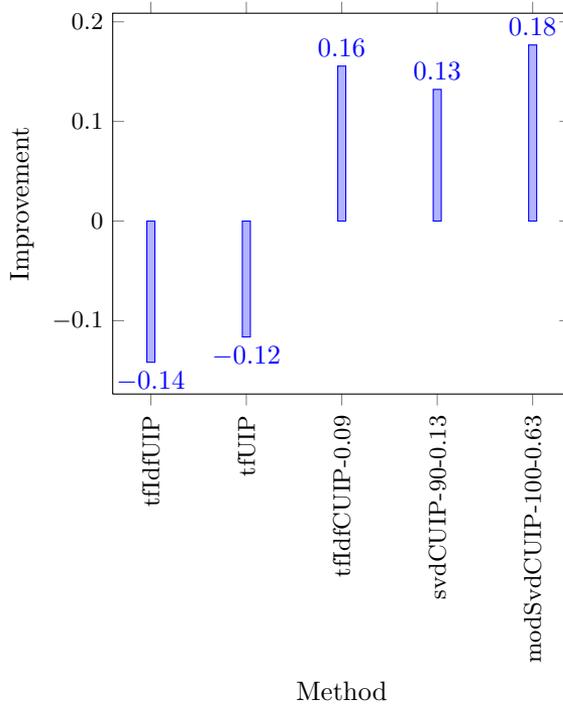


Figure 4.12: Comparing the Improvement of $tfIdfUIP$, $tfUIP$, $tfIdfCUIP-0.09$, $svdCUIP-90-0.13$, $modSvdCUIP-100-0.63$

performance. The best performance of $modSvdCUIP$ for the AOL query data set was observed when the dimension parameter k was set to 100. The average document space for the AOL data set is 500, significantly more than that of the custom data set. These results shows that the $modSvdCUIP$ was benefited from the dimensional reduction step. The $svdCUIP$ based personalized search also benefits from the dimension reduction step. For both data sets, the best performance was achieved when k was set to 90. We can draw the conclusion

4.4 Experimental Evaluation

that both approaches profited from the dimension reduction step. However, due to some small values in the similarity matrix Sim , the HAC algorithm couldn't clearly distinguish clusters that resulted in many small-size clusters, i.e., a topic is divided among several clusters. This resulted in poor performance of *svdCUIP* based personalized search compared to *modSvdCUIP* based personalized search in which the *modSim* matrix has comparatively higher similarity values, enabling HAC to clearly distinguish the clusters.

What distinguishes *CUIP* based personalized search approaches with other works that use social bookmarking services for personalized search is that tags in a user's *UIP* are dealt locally, and tags that constitute a *CUIP* are part of the vocabulary of a community of users who have annotated the documents clicked by the user. Tags in a user's *UIP* constructed based on (Noll and Meinel, 2007), (Xu, Bao, Fei, Su, and Yu, 2008), and (David, Iván, and Joemon, 2010) approaches are those used by the user to annotate documents of interest. As mentioned in the related work, there is a discrepancy between the vocabulary a user sees to formulate a search query and the vocabulary used in Web documents. Using only the user vocabulary to construct a *UIP* suffers from incomplete, insufficient tags. Building a user's *UIP* with tags that encompasses the world view can surpass this limitation to a certain extent.

(Noll and Meinel, 2007) doesn't include user and resource length normalization factor in the computation of cosine similarity score formulae. They neither normalize user profile tag frequencies nor resource profile tag frequencies; the tag weight of tags in the *UIP* is calculated by accumulating the count of tags,

4.4 Experimental Evaluation

and the term weight of terms in the resource profile is set to 1, if the term is used for annotating a document, else 0. This would allow equal importance to all documents and to all users. It makes sense not to normalize the tag weight of tags in user profile, because the terms were those that the user scribbled him (her)-self to annotate the documents. Xu's et al, on the other hand, use user and document length normalization factors resulting in the degradation of personalized search performance. Vallet et al. follows the same philosophy of Noll et al, and they adapt the Xu's approach by eliminating the user and document length normalization factor. Their justification for exclusion of normalization factor is similar to Noll's work that using the document length normalization factor would penalize the score of popular documents. Note that, similar to Noll's work, their approach also use all the tags in the *UIP* to compute the similarity score for re-ranking documents. Also, the similarity function computes the vector product of $t_{fu} * i_{uf}$ and $t_{fd} * i_{df}$ to calculate the similarity between *UIP* and document, where t_{fu} , t_{fd} , i_{uf} , and i_{df} is term frequency of a term in user profile, term frequency of a term in document profile, inverse user frequency, and inverse document frequency, respectively. Again, this kind of computation is only possible if we assume that every user who is searching the Web, (s)he is also actively tagging documents, otherwise how would one calculate i_{uf} . We present a more realistic approach, achieving a little better performance than (David et al., 2010), and making no assumptions about user's tagging activity. (Andriy, Jonathan, Bamshad, and Robin, 2008) presented a personalization algorithm for recommendation in folksonomies which relies on hierarchical tag

4.4 Experimental Evaluation

clusters. Note that the work is not about personalized search, but an adaptation of personalized search for recommendation of resources to the users of the folksonomy system based on their previous annotation of resources. Their approach clusters the entire tag space of the folksonomy system to obtain a common cluster structure to be used by all users of the folksonomy system. This approach is only applicable in a folksonomy system. Given a common cluster structure, tags in a user's *UIP* are mapped to appropriate clusters. It is like mapping a list of tags that have local scope to tag clusters that have global scope. This will augment the tags in the user *UIP*, thus encompassing the user's own vocabulary and of the community. A cluster structure will have all the possible semantic terms related to a topic. For example: consider a user's *UIP* has tags related to religion such as jewish, Israel, religion, etc (local scope). These tags will be mapped to a cluster that has the topic 'religion' in the common cluster structure (global scope). The mapped cluster may also have other tags related to religion such as Hinduism, Christianity, Buddhism, etc. Such kind of CUIP has properly identified the user general interests, for example, religion in this scenario, but it fails to identify the user's specific interests, which was originally jewish, but now after the CUIP is augmented, it also contains additional terms such as hinduism, buddhism, etc. To circumvent this limitation, (Andriy, Jonathan, Bamshad, and Robin, 2008) proposed to use three tuning parameters, step, generalization level, and division level, to limit the breadth of the mapped cluster. Our approaches also try to achieve the same objective, which is user oriented and bounded by the tags in the user's

4.4 Experimental Evaluation

UIP to generate a *CUIP*. However, we don't need any special parameters to limit the breadth of the cluster structure. This reduces the complexity and maximizes the accuracy of computing the cluster structure, also also increases the search quality. We also observed that not all queries benefit from the personalized search; the self-evident queries, also referred as navigational query (Broder, 2002), need not always be disambiguated, because the target Web documents for these queries are the same regardless of user interest. We found that applying personalized search to navigational queries reduces performance. The vague queries, which need to be disambiguated or could have different answers depending on user interests, benefit from the application of *CUIP* based personalized search.

One limitation of our proposed methods is that both the *UIP* and *CUIP* depend on the resource profile of resources. Tags in a *UIP* are aggregated from the resource profiles of Web documents. A resource profile for a Web document is only available if its annotations are available on a public social bookmarking service. We found for the AOL data set that approximately 34% of all the Web documents were not annotated on Delicious servers. Whereas, for the custom data set, 45% were not annotated on the same servers. One reason for this difference lies in the age of data sets: the AOL data set is older, hence there is a higher probability of the data being annotated. In our future work, we would like to experiment with OpenCalais¹ service for Web documents whose resource profiles cannot be constructed from social bookmarking services. OpenCalais

¹www.opencalais.com

4.4 Experimental Evaluation

returns topics, place names, people names, and URLs present in a document. This will also help us to develop a much better *UIP* and to improve the quality of personalized search.

Finally, the proposed methods can be used for personalizing search results generated from any search engines, and are very compatible for building a *UIP* or *CUIP* from any social bookmarking services. Our key contribution rests in developing a *CUIP*, and showing its usage for personalized search, one of many areas our methods can be applied for.

To conclude, the cluster structure emerging from a *modSvdCUIP* is able to identify user interests, group semantically related tags into clusters, identify second-order co-occurrence similarity between terms, and improve the search result quality. Personalized search based on *modSvdCUIP* performs better than approaches using the *tfUIP*, *tfIdfUIP*, and is comparable to the approaches *tfIdfCUIP* and *svdCUIP*. The improvement is due to the fact that the similarity matrix *modSim* is able to discover the sense of a topic by computing the first-order co-occurrence and second-order co-occurrence similarity between tags.

5

User Profiling for Partnership Match

In order to maximize the advantages and minimize the negative effects of globalization and growing interdependence, it is imperative for SMEs (Small and Medium Enterprises) in developing countries to forge partnerships with big enterprises in developed regions. However, the partnership establishment process is a rough ride; it comes with its own set of hurdles. A survey by PricewaterhouseCoopers (PwC) reveals that 44% of the partnerships were unsuccessful. We refer to research literature to find out various features that are involved during partnership establishment process. Based upon a review, we select features that form core concepts in a partnership establishment process. These concepts along with their related properties are modeled as an ontology, termed as Partnership Ontology. Big enterprises and SME (Small and Medium Enterprises) can use the partnership ontology to lay down their requirements as

a buyer profiles and seller profiles respectively. A semantic similarity measure is defined to compute a ranked list of matching seller profiles given a buyer profile. We illustrate the devised methodology of partnership establishment process by an example using a case study.

Globalization has ushered new gateways for SMEs in developing countries through greater integration into the world economy. The possibility to import new ideas, modern technology, and business investment opportunities from advanced countries can boost economic growth. Significant transfer of technology and modernization of the economies has occurred particularly in manufactured goods, through joint ventures, licensing agreements and other enterprise partnerships. Partnership is a voluntary collaborative agreement between two or more parties in which all participants agree to work together to achieve a common purpose or undertake a specific task which is a win-win situation for both. PwC interviewed CEOs of 239 Fortune 500 companies - results show that 56% of the companies in US have partnered over the past 5 years. These companies have partnered with large companies (41%), large MNCs (28%), large domestic companies (22%), small companies (29%), university (7%), and federal lab (3%).

A common theme among purchase managers from both failed and successful strategic alliances is the importance of building mutual trust and commitment among partners. No matter how mutually beneficial and logical the venture may seem without trust and commitment the alliance will fail entirely, or it will fail to reach its strategic potential. There are a variety of ways that a

company can attain and sustain commitment and trust in cooperative ventures. Goal and intent revelation is a crucial step toward building trust. The most common causes of failure ¹ cited by CEOs are: cultural differences (49%), poor or unclear leadership (49%) and poor integration process (46%). Though most enterprises understand and are aware of the reasons of the failure, they somehow fail to establish an amenable partnership. This is because they fail to spend enough resources understanding their individual needs and defining their requirements. As a result, there is a greater risk of an incorrect decision that ultimately leads to failed relationships

The projects that operate within inter-enterprise environments additionally face the problem that different information models are likely to be used by different partners. Engineers working within a particular organization will inevitably develop their own vocabulary for particular activities and these will need to be adjusted to be more practical and to meet the requirements of different collaborating partners. Hence, when two different partners are brought together, two common types of problem can occur in communications that share and exchange information, firstly, the same term is being applied to different concepts (semantic problem), secondly, different terms may be used to denote the same entity (syntax problem). This problem is popularly known as integration problem (Giachetti, 2004) in literature. Employment of ontology in this work resolves the integration problem. Thus a critical question is, how geographically separated organizations can be supported to establish a part-

¹http://www.1000ventures.com/business_guide/partnerships_main.html

nership that increases the probability of success?

In the previous two chapters, chapter 3 and chapter 4, I have presented how feature based user profiling can be used for building *UIPs* and *CUIPs*. In chapter 3, the feature anchor text of clicked Web pages by the user was used for building *UIPs*. In chapter 4, the feature tag annotations by a community of users to the clicked Web pages by the user was used for building *UIPs* and *CUIPs*. In this chapter, the features that are targeted are user preferences and context of work, refer Figure 1.1. A user explicitly input his preferences (attribute values) about the attributes of interest. Attributes are predefined and modelled as concepts in an ontology representing the context of work. This chapter also demonstrates how a buyer profile or seller profile is constructed by explicitly requesting a user to input his preferences about the concepts defined in the ontology, and how similarity is computed between different types of profiles. This chapter makes the following contributions:

1. I survey the research literature to identify the key concepts that are negotiated during a partnership establishment process.
2. Based upon the concepts identified in the previous step, an ontology is proposed, termed as Partnership Ontology.
3. Using Partnership Ontology, a manifestation of user profiles is illustrated as buyer profiles or seller profiles.
4. A semantic similarity match is proposed that recommends matching seller profiles for a given buyer profile.

5.1 Supplier Selection

In the traditional Supplier Selection process, an enterprise scrutinize potential suppliers from a given list of suppliers. An enterprise select potential suppliers from its previous dealings. A RFQ (Request For Quotation) is sent to all the potential suppliers. After receiving quotes from suppliers and based on the various other information listed in Table 5.1, an optimal supplier is selected. The whole process of supplier selection can be summarized into 6 steps:

1. Select Candidate Suppliers
2. Send RFQ (Request for Quotation)
3. Receive Quotations
4. Select Supplier
5. Negotiation
6. Signing the Contract

Though the above 6 step process for Supplier Selection looks trivial, it is a very time consuming and complex process. We list the various complexities that one encounters and side by side explain how our system deal with them.

1. To select potential suppliers, a buyer use the previous history or its dealings with the suppliers. This limits the number of supplier and hence

5.1 Supplier Selection

lower the competitiveness of the supplier selection process. New suppliers, who have had no interaction with the current buyer but have successful partnerships with other buyers, are not given due consideration. In-order to remove any biases, our system allows all suppliers to model their facilities or services as a seller profile.

2. Sending RFQ and receiving quotations is a time consuming process. Moreover RFQs are best suited to standardized products or services so that various supplier quotes can be easily comparable. This is a serious limitation which limits a system applicable to only a particular domain. The proposed system uses UNSPSC ontology ¹ for disambiguation of any product or services. The UNSPSC provides an open, global multi-sector standard for efficient and accurate classification of products and services. Using UNSPSC codes throughout an extended supply chain - seller, buyer, and distributor can process transaction data automatically and can perform management, analysis and decision function in time-critical ways that would not be possible without the codes. Classifying products and services with a common coding scheme facilitates commerce between buyers and sellers and is becoming mandatory in the new era of electronic commerce. Large companies are beginning to code purchases in order to analyze their spending. Classifying products and services with a common coding scheme facilitates commerce between buyers and sellers and is becoming mandatory in the new era of electronic

¹<http://www.ksl.stanford.edu/projects/DAML/UNSPSC.daml>

5.1 Supplier Selection

commerce. Large companies are beginning to code purchases in order to analyze their spending. The UNSPSC is designed to serve three primary functions: **Resource Discovery**, **Expenditure Analysis**, and **Product Awareness**. UNSPSC is a hierarchical classification having 5 levels, altogether it is a eight or ten digit numerical code. The codes are hierarchical, similar to an outline. As you get deeper in the outline, there is more detail. Each level contains a two character numerical value and a textual description. Based on this hierarchical structure, each UNSPSC code can be broken down as follows: the first 2 digits (from left) represent *segment*, next 2 digits represent *family*, next 2 digits represent *class*, second last 2 digits represent *commodity* and finally the last 2 digits are optional that represent *business function*. For ex., the UNSPSC code for Cooling or refrigeration services is 70142011 which is comprised of following categories. The *segment* code 70 for “Farming and Fishing and Forestry and Wildlife Contracting Services”, *family* code 14 for “Crop production and management and protection ”, *class* code 20 for “Post harvesting crop processing”, and finally the *commodity* code 11 for “Cooling or refrigeration services”.

3. An RFQ typical involves listing detailed specification of products or services. The more detailed the specifications, the more accurate the quote will be and comparable to the other suppliers. There is no standard for unit of measure and no distinct identifier for different product packaging

5.2 Criteria for Partnership Establishment

levels. For instance , one may order 20 and receive 200 because they are sold in units of 10. This results in inventories of wrong products and increased returns processing, driving up costs and creating cash flow issues. This work proposes a partnership ontology, that models the specifications as features and properties, also models unit of measurements similar to GoodRelations Ontology, refer (Hepp, 2008). Table 5.1 provides a snapshot of some of the important features that plays a key role for buyer - Supplier decisions are typically made following a comparison and analysis of the features.

5.2 Criteria for Partnership Establishment

The focus of work in this chapter provides a framework for establishment of buyer-seller partnership, where buyer are big enterprises and suppliers are SME (Small and Medium Enterprises). This section, in particular, investigates the core features or concepts required for building a profile i.e. the final goal results in a set of concepts and related properties that form an ontology for partnership establishment. The success of an establishment process is greatly reduced with the requirements criteria and their associated attributes being clearly known before the evaluation approach is implemented. In software engineering, requirement analysis encompasses those tasks that go into determining the needs of a customer. Requirement analysis determines the set of criteria to identify business needs i.e. what one party hopes to attain from another. The

5.2 Criteria for Partnership Establishment

complex process of partnership establishment generally involves assessing multiple criteria of varying importance, which may be quantitative or qualitative, tangible or intangible and which may involve trade-offs. (Dickson, 1966) and (Weber, Current, and Benton, 1991) provides a list of criteria that SMEs or enterprise negotiate over. Some of these criteria have gone obsolete over time due to changing business needs; therefore, we augment this list according to current requirements of partnership establishment process, refer Table 5.1. For example consider a scenario where a partnership under consideration between two geographically separated organizations, say one in USA and other one in Vietnam. Both partners have a different motivation for forging a partnership; an SME in Vietnam may be interested in a partnership so that they could learn advance technology whereas an organization in USA may be interested because of cheap labor costs. Since their motivations are different their requirements must also be different. Some of the other important criteria are discussed below. Financial Stability is one of the core requirements of a buyer; a SME with lot of debts can run the project into trouble. A match much be drawn between buyer requirements and seller manufacturing skills. Research and Development R&D includes assessing a potential partners level of R&D investment, the number of personnel involved in R&D, the communication network in place, the skill level of R&D personnel, and whether or not the organization engages in developing new products, and product and process improvement. A strong R&D presence in a potential partner organization is a positive sign for partnership. The next criterion is market knowledge and marketing skills, which involves

assessing the potential partners' market presence and understanding of both their competitors and customers. Alignment between the cultures of the SMEs and potential partner includes examining the cultural understanding between both organizations and their individual practices and behavior. A partnership often involves give and take or learning from each other, the willingness to share expertise criteria captures the notion of compatibility. One of the major criteria for forging partnership is trust which can be modeled using previous alliance experience. However, we strongly feel that trust should have more concrete concepts, therefore we have added more concepts under trust to model it comprehensively.

5.3 Partnership Ontology

In the following, we give an overview of the relevant conceptual entities and types of relationships. A definition of ontology by (Fensel) describes it as "specifically machine-readable information whose meaning is well defined by standards, which absolutely needs the inter-operable infrastructure that only global standard protocols can provide". The concept involves categorizing structured and semi-structured information in a standard manner in order to give it meaning so that machines can understand it, process it and hence derive additional information, if any. Partnership ontology in Figure 5.1 is formulated from the concepts in Table 5.1; explained below are some additional concepts and properties that explain the relationship between them.

5.3 Partnership Ontology

Table 5.1: List of Concepts produced by amalgamating contribution of various research work's in domain of Partnership Establishment.

Research Articles	
(Chen, Lee, and Wu, 2008)	x
(Hans, 2008)	x
(Wang and Kess, 2006)	x
(Bayazit, 2006)	x
(Maheshwari, Kumar, and Kumar, 2006)	x
(Pidduck, 2006)	x
(Shekhar, 2008)	x
(Choy and Lee, 2003)	x
(Jagersma and van Gorp, 2003)	x
(Lemke, Goffin, and Szwedjczewski, 2003)	x
(Kaplan and Hurd, 2002)	x
(Dyer, 2000)	x
(Dacin, Hitt, and Levitas, 1997)	x
(Lin and Hai, 2005)	x

Financial Stability	current profits	x
	growth potential	x
	cash flow	x
	equity	x
	debt amount	x
Unique Competency		x
Capability Compatibility	manufacturing skills	x
	manufacturing facilities	x
	human resources	x
	technology R&D	x
	management	x
	quality control	x
	collaboration	x
	future capabilities	x
Market Attractiveness	Market Knowledge	x
	Marketing Skills	x
	Market Share	x
	Marketing Objectives	x

5.3 Partnership Ontology

In-order to build a common terminology for both enterprises and SMEs most of the concepts are modeled as enumeration. For ex: the concept **currency** is modeled as enumeration with two values USD and EURO; thus any concept that link to currency can only use USD and EURO as values. The partnership ontology is centered around concept **Profile**. Every **Business Entity** that wish to use this ontology must define a **Profile**. A **Profile** can be either **Buyer Profile** or **Seller Profile**. A concept **Profile** is modeled as a super concept of concept **Buyer Profile** and **Seller Profile** and all the properties are defined on concept **Profile**. Because of entailment rules, all the properties defined on concept **Profile** are inherited by both sup concepts **Buyer Profile** and **Seller Profile**. The concept **Profile** has properties that are instrumental in defining profiles; for ex: properties *financialStatusisDefinedBy*, *hasCapability*, *hasCoreCompetency* can define a user's profile financial status, manufacturing skills, manufacturing units and core competency respectively. Every profile has a validity duration which is modeled using two data type properties *validFrom* and *validThrough*.

The concept **FinancialStability** uses the concept **AnnualReport** to define an enterprise financial conditions and both concepts are related together using the property *annualReport*. The concept **AnnualReport** define various properties that can help where the annual report document can be located (*avaiableAt foaf:Document*), how much is the debt amount(*hasDeptAmount*), how much is the liquidity(*hasEquityAmount*), how much is the cash flow (*hasCashFlow*).

5.3 Partnership Ontology

The concept **Capability**, defines the core strength of an organization, is a super concept of three concepts **ManufacturingCapability**, **MarketCapability**, and **TechnologyRnD**. Note that, concepts **Manufacturing Skills** and **Manufacturing Facility** are enumerations. To model trust, which is a very essential part in any partnership establishment, we use the past history of alliances. A SME is trustworthy if he/she has successfully executed projects in partnership with other enterprises. Therefore, the concept **Trust** has a property *partners* which connect to concept **PartnerList**. Using the concept **PartnerList**, a number of partners can be defined, and each partner is modeled using the concept **Partner**. A partner is identified using the properties *foaf:homepage* and *foaf:name* to name a few. A concept **Partner** also contains information about domain of alliance modeled using property *unitOfAlliance* connected to concept **ManufacturingUnit** which can be further narrowed down to a particular manufacturing skills using the property *hasDomain*. The range of property *hasDomain* is **ManufacturingSkills** which represent the core service area. There can be various approaches to modeling **Manufacturing Skills**. The simplest approach could be instances of concept **Manufacturing Skills** be string literal which can create disambiguation, for ex: if a user uses a string value "Refrigeration", this has several further variations like "Industrial Refrigeration", "Cooling and Refrigeration Services", etc. It may be possible that engineers working at different organizations have different vocabulary - this would seriously effect the similarity match results of profiles. We propose to use UNSPSC web service, as described in section 3,

for disambiguation of **ManufacturingSkills**. Given a string literal, our system search its matching standard terms in the UNSPSC and return them in order of relevance. For ex: for string literal refrigeration, four matching terms are returned “Industrial refrigeration ”, “Cooling or refrigeration services ”, “HVAC refrigeration construction service ”, and “Air conditioning or ventilating or refrigeration equipment manufacture services”. Note that UNSPSC also returns the unique UNSPSC codes for each of the term. These standard codes are stored as an instance of **ManufacturingSkills**. Each manufacturing unit also contains information about risk assessment i.e. if an enterprise has implementation of risk assessment guidelines in their factory or workplace.

Another important concept for forging partnerships is partner marketing skills. This is modeled using the concept **MarketCapability** which is related to concept **Profile** using the property *hasMarketCapability*. The concept **MarketCapability** models the market skills and market knowledge of an SME using the properties *hasMarketSkills* and *marketKnowledge* respectively which are further related to enumerated concepts **Market Share** and **MarketSkills**. Concepts and Sub-concepts henceforth will be referred to as attributes and concept instances will be referred as attribute values.

5.4 Case Study

Most of the research work in the domain of Partnership Establishment takes a manual approach; asking purchase managers who participate in the study to

evaluate suppliers on a set of features and some sort of scale. It is important to note that, such a study only provides a subjective view of a set of managers and it would be inappropriate if their evaluation be generalized for the whole population. Therefore, the work in this chapter takes a personalized view - we ask the suppliers or sellers and buyers to provide their information and services respectively as a profile. We evaluate five candidate suppliers and one buyer using partnership ontology and semantic similarity measure. One Buyer profile and five supplier profiles are shown Figure 5.2, 5.3, 5.4, and 5.4. The information about suppliers and buyers were provided by the **Trade Investment Agency** (name withheld due to privacy issues). The provided information was then represented using partnership ontology.

5.4 Case Study

Name: seller1		Name: seller2	
Current Profits	100K-999K	Current Profits	1-99K
Manufacturing Facilities	ISO 6 class clean room	Manufacturing Facilities	ISO 6 class clean room
Growth Potential	> 1 Million	Growth Potential	
Cash Flow	100K-999K	Cash Flow	100K-999K
Human Resources	> 500 employees	Human Resources	100-499 employees
Currency	Euro	Currency	USD
Equity	100K-999K	Equity	
Technology R&D	Information management	Technology R&D	Information management
Debt Amount	1-99K	Debt Amount	
Unique Competency	Automotive manufacturing	Unique Competency	Automotive manufacturing
Management	greater 10 years management experience	Management	
Manufacturing Skills	Electronics	Manufacturing Skills	
Quality Control	ISO 14001:2004 certification	Quality Control	ISO 14001:2004 certification
Collaboration		Collaboration	
Future Capabilities		Future Capabilities	
Market Knowledge	Good market knowledge	Market Knowledge	
Marketing Skills	Extensive marketing skills	Marketing Skills	Good marketing skills
Market Share	20 - 49% market share	Market Share	
Marketing Objectives		Marketing Objectives	
Market Gaps	luxury cars	Market Gaps	
Partnership Potential		Partnership Potential	
Trust		Trust	
Personal Rapport		Personal Rapport	
Commitment		Commitment	
Reputation		Reputation	
Dependancy		Dependancy	
Flexibility		Flexibility	
Cultural Alignment		Cultural Alignment	High level cultural alignment
Willingness to Share Expertise	High level willingness	Willingness to Share Expertise	
Previous Alliance Expertise	Yes	Previous Alliance Expertise	
Partnership Strategy		Partnership Strategy	
Shared Goals		Shared Goals	
Location		Location	
Political Links		Political Links	
Risk Assessment		Risk Assessment	
Relationship Maintenance		Relationship Maintenance	

Figure 5.2: Seller Profiles for this study: Seller1 and Seller2

5.4 Case Study

Name: seller3		Name: seller4	
Current Profits	1-99K	Current Profits	1-99K
Manufacturing Facilities	ISO 6 class clean room	Manufacturing Facilities	ISO 6 class clean room
Growth Potential	100K-999K	Growth Potential	100K-999K
Cash Flow	100K-999K	Cash Flow	100K-999K
Human Resources	50-99 employees	Human Resources	50-99 employees
Currency		Currency	USD
Equity	100K-999K	Equity	100K-999K
Technology R&D	CAD	Technology R&D	CAD
Debt Amount	100K-999K	Debt Amount	100K-999K
Unique Competency	Automotive manufacturing	Unique Competency	Automotive manufacturing
Management		Management	1 - 4 years management experience
Manufacturing Skills	Electronics	Manufacturing Skills	
Quality Control	ISO 9001:2000 certification	Quality Control	ISO 9001:2000 certification
Collaboration		Collaboration	
Future Capabilities		Future Capabilities	
Market Knowledge		Market Knowledge	
Marketing Skills	Limited market skills	Marketing Skills	
Market Share		Market Share	
Marketing Objectives		Marketing Objectives	
Market Gaps		Market Gaps	
Partnership Potential		Partnership Potential	
Trust		Trust	
Personal Rapport		Personal Rapport	
Commitment		Commitment	
Reputation		Reputation	
Dependency		Dependency	
Flexibility		Flexibility	
Cultural Alignment		Cultural Alignment	
Willingness to Share Expertise	High level willingness	Willingness to Share Expertise	High level willingness
Previous Alliance Expertise		Previous Alliance Expertise	Yes
Partnership Strategy		Partnership Strategy	
Shared Goals		Shared Goals	
Location		Location	Korea
Political Links		Political Links	
Risk Assessment		Risk Assessment	
Relationship Maintenance		Relationship Maintenance	

Figure 5.3: Seller Profiles for this study: Seller3 and Seller4

5.4 Case Study

Name: seller5	
Current Profits	1-99K
Manufacturing Facilities	10-499 sqm
Growth Potential	
Cash Flow	1-99K
Human Resources	
Currency	Euro
Equity	1-99K
Technology R&D	Robotics
Debt Amount	
Unique Competency	Automotive manufacturing
Management	
Manufacturing Skills	Welding
Quality Control	ISO 9001:2000 certification
Collaboration	
Future Capabilities	
Market Knowledge	
Marketing Skills	
Market Share	
Marketing Objectives	
Market Gaps	
Partnership Potential	
Trust	
Personal Rapport	
Commitment	
Reputation	
Dependency	
Flexibility	
Cultural Alignment	
Willingness to Share Expertise	
Previous Alliance Expertise	
Partnership Strategy	
Shared Goals	
Location	
Political Links	
Risk Assessment	

Figure 5.4: Seller Profiles for this study: Seller5

Attribute Manager

Request Name : AutomotiveRequest

Attribute Library	Attribute Values
<ul style="list-style-type: none"> Current Profits Manufacturing Facilities Growth Potential Cash Flow Human Resources Cash Flow Equity Technology R&D Debt Amount Unique Competency Management Manufacturing Skills Quality Control Collaboration Future Capabilities Market Knowledge Marketing Skills Market Share Marketing Objectives Market Gaps Partnership Potential Trust Personal Rapport Commitment Reputation Dependency Flexibility Cultural Alignment Willingness to Share Expertise Previous Alliance Expertise Partnership Strategy Shared Goals Location Political Links Risk Assessment Relationship Maintenance 	<div style="border: 1px solid black; padding: 5px;"> <p>Marketing Skills Good marketing skills</p> <p>Risk Assessment Medium risk</p> <p>Partnership Potential Good partnership potential</p> <p>Technology R CAD</p> <p>Equity > 1 Million</p> <p>Cash Flow 100K-999K</p> <p>Human Resources > 500 employees</p> <p>Manufacturing Facilities ISO 6 class clean room</p> <p>Current Profits 100K-999K</p> <p>Willingness to Share Expertise High level willingness</p> <p>save</p> </div>

Figure 5.5: An example to demonstrate construction of user profile (Buyer Profile) - concepts shown here are derived from the Partnership Ontology

5.4.1 Buyer Profile and Seller Profile

The success of partnership establishment is significantly influenced by the manner in which profiles are constructed. A profile is simply a set of generic facts about a company, which may be used by other companies to determine their suitability as potential partners. A seller profile is a mechanism utilized to communicate what the potential partner can do to meet their needs. A seller profile records the capabilities and services that he has for offer. A buyer profile is a mechanism utilized to communicate the expectations that an enterprise has from a potential partner. Both the profiles are generated using the Partnership Ontology introduced in Section 5. An enterprise (henceforth called as buyer) looking for partners makes a buyer profile, whereas, SMEs make a seller profile. Note that both are oblivious of each other, i.e. they just make their profiles available to the system. Buyer, after providing his profile to the system, searches for the matching seller profiles, which the system returns after executing a semantic similarity match among various seller profiles available to the system. The result from searching is a set of possible partners that a buyer can consider to be his/her future partners. We developed a web service, that uses Partnership Ontology to construct seller profile and buyer profile using the Partnership Ontology, termed as e-Partner. This web service is developed using Java technologies, AJAX, Java Script and HTML. The web-service is available on-line and accessible through the following URL <http://tinyurl.com/yau5mfg>. Figure 5.6 and Figure 5.5 shows an exemplary use of web service to create a

Buyer Profile or Seller Profile.

After building a buyer profile, an enterprise can search for matching seller profiles by using the search functionality. But, before using the search option, a buyer can set the weights for the attributes which associates importance to the attributes, refer Figure 5.6. The weight assigned to attributes signifies the importance of the attribute and is used in the calculation of similarity distance i.e. if a particular attribute in a buyer profile has weight 0.5 and the same attribute is also present in a seller profile, its similarity score will be greater, however if it is absent in a seller profile then similarity score for that particular attribute will be 0. The knock-out property selected for a particular attribute in a buyer profile can be interpreted as follows; if a seller profile does not has that attribute in its profile, simply discard the profile. In other words, knock out property makes an attribute essential and puts a restriction that a prospective seller has to have that attribute in its profile. A sourcing property for a particular attribute if checked signifies that this particular attribute is insignificant. In other words, if an attribute, is checked for sourcing property in a buyer profile and, is missing from a seller profile, it will still be considered for calculating the overall similarity score. For instance, if a buyer profile has 3 attributes a_1 , a_2 , and a_3 , and a seller profile has 2 attributes a_1 and a_2 , this evaluates to 66.67% similarity, but, if a buyer profile has the sourcing property selected for a_3 , similarity score will now evaluate to 100%. Note that similarity score of any 2 attributes also depends on the depth of attribute values. The sourcing property is included for experimentation, so that a buyer can actually

evaluate how many sellers show up if they unselect a particular attribute. Also note that, weight, sourcing and knock-out properties are not available for a seller profile.

5.4.2 Semantic Similarity Measure

Given a collection of buyer profiles and seller profiles, the next step would be to find a ranked list of seller profiles for a given buyer profile. In order to compute a ranked list, we propose a semantic similarity measure which is motivated from (Salton, Wong, and Yang, 1975) work on Vector Space Model. First, we briefly explain what is vector space model and how it can be modelled to suit our needs. Following it, we postulate two definitions to lay the basis for mathematical formulate for computation of similarity measure of profiles.

5.4 Case Study

Name: AutomotiveRequest				
Competencies	Value	Weight	Sourcing	Knockout
<u>Current Profits</u>	100K-999K	0.5	<input type="checkbox"/>	<input type="checkbox"/>
<u>Manufacturing Facilities</u>	ISO 6 class clean room	0.2	<input type="checkbox"/>	<input type="checkbox"/>
<u>Growth Potential</u>		0.1	<input type="checkbox"/>	<input type="checkbox"/>
<u>Cash Flow</u>	100K-999K	1	<input type="checkbox"/>	<input type="checkbox"/>
<u>Human Resources</u>	> 500 employees	1	<input type="checkbox"/>	<input type="checkbox"/>
<u>Currency</u>	USD	0.1	<input type="checkbox"/>	<input type="checkbox"/>
<u>Equity</u>	> 1 Million	0.8	<input type="checkbox"/>	<input type="checkbox"/>
<u>Technology R&D</u>	Information management	0.1	<input type="checkbox"/>	<input type="checkbox"/>
<u>Debt Amount</u>	1-99K	0.8	<input type="checkbox"/>	<input type="checkbox"/>
<u>Unique Competency</u>	Automotive manufacturing	1	<input type="checkbox"/>	<input checked="" type="checkbox"/>
<u>Management</u>	High level management support	0.5	<input type="checkbox"/>	<input type="checkbox"/>
<u>Manufacturing Skills</u>	CNC machining	1	<input type="checkbox"/>	<input type="checkbox"/>
<u>Quality Control</u>	ISO 14001:2004 certification	0.8	<input type="checkbox"/>	<input type="checkbox"/>
<u>Collaboration</u>		0.1	<input type="checkbox"/>	<input type="checkbox"/>
<u>Future Capabilities</u>		0.1	<input type="checkbox"/>	<input type="checkbox"/>
<u>Marketing Skills</u>	Good marketing skills	0.6	<input type="checkbox"/>	<input type="checkbox"/>
<u>Market Share</u>	20 - 49% market share	0.6	<input type="checkbox"/>	<input type="checkbox"/>
<u>Cultural Alignment</u>	High level cultural alignment	0.1	<input type="checkbox"/>	<input type="checkbox"/>
<u>Willingness to Share Expertise</u>	Medium level willingness	0.1	<input type="checkbox"/>	<input type="checkbox"/>

Figure 5.6: A reduced version of buyer profile - truncated to fit in here. The features that buyer does not choose during profile construction are removed to save space.

VSM is a linear algebraic method most commonly used in Information Retrieval for representing text documents as vectors and aids in relevancy ranking of documents with respect to the inputted query. A document is represented as a vector in an m dimension subspace, where m constitutes the number of words in the dictionary. If a word or term occurs in the document, its value in the vector is 1 otherwise 0. Hence, such kind of vector tends to be sparse. Moreover, if we constitute a term-document matrix i.e. terms as rows and documents as columns, the matrix formed will be sparse matrix. Motivated by the terminology used in Vector Space Model, we would like to borrow it, improvise it and use it in the context of supplier match. Here, we define a *profile vector* and an *attribute-profile matrix* to suit Vector Space Model to our needs. The profile-attribute matrix will not be very high dimensional because in the current scenario attributes are finite as compared to terms in a dictionary which are infinite (or a very large number).

Definition 1: A *Profile Vector* $P^{(i)}$ is represented by a m -dimensional vector

$$P^{(i)} = \{att_1, att_2, \dots, att_m\} \quad (5.1)$$

where att_m , is a name of an attribute.

The actual *Profile Vector* P^i after substitution of values for attributes will be

$$P^{(i)} = \{av_{i1}, av_{i2}, \dots, av_{im}\} \quad (5.2)$$

where av_{im} is a value for att_m for profile $P^{(i)}$.

Definition 2: An *Attribute-Profile Matrix* is a mathematical matrix that describes the value of various attributes that occurs in a collection of profiles. Each column correspond to a profile in the collection, and each row corresponds to an attribute with its attribute-value.

$$A_{n,m} = \begin{pmatrix} av_{1,1} & av_{2,1} & \cdots & a_{n,1} \\ av_{1,2} & a_{2,2} & \cdots & a_{n,2} \\ \vdots & \vdots & \ddots & \vdots \\ a_{1,m} & a_{2,m} & \cdots & a_{n,m} \end{pmatrix} \quad (5.3)$$

Now, a column in the *Attribute-Profile Matrix* is a column vector corresponding to a profile, giving its relation to each attribute.

Given the profile vectors for two different profiles (of course, one is a buyer profile and other is a seller profile), it is possible to compute a similarity between them, $sim(P^i, P^j)$, which reflects the degree of similarity between two profiles. Such a similarity measure will be an inner product of the two vectors. When two vectors are identical, the cosine of angle between them will be 0, producing a maximum similarity.

Suppose, let us represent an exemplary profile vector according to definition 1 as $\{P^{(i)}; i=1, \dots, n\}$ of attributes of n different partners. A profile vector, $P^{(i)}$, will be represented in m -dimension subspace as a vector, where m -dimension subspace consists of m different attributes represented in space. Equation 4

shows a $1 \times m$ column matrix representation of profile vector (buyer or seller).

$$P^{(i)} = \begin{bmatrix} att_1 \\ att_2 \\ \vdots \\ att_m \end{bmatrix} \quad (5.4)$$

Or, a profile with attribute-values substituted for attributes will be

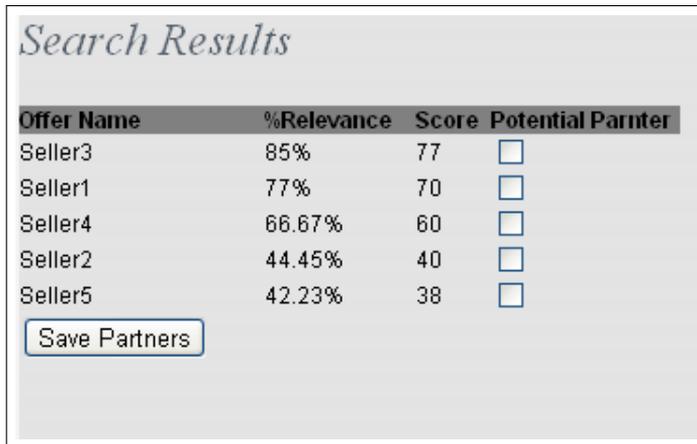
$$P^{(i)} = \begin{bmatrix} av_{i1} \\ av_{i2} \\ \vdots \\ av_{im} \end{bmatrix} \quad (5.5)$$

To compute the similarity of a buyer profile with seller profiles, we can take cross product of vector representation of buyer profile with various seller profiles using equation 6.

$$Sim(P^{(i)}, P^{(j)}) = \frac{\sum_{k=1}^m av_{ik} * av_{jk}}{\sqrt{\sum_{k=1}^m av_{ik}^2} * \sqrt{\sum_{k=1}^m av_{jk}^2}} \quad (5.6)$$

Equation 6 aids in generating a ranked list of seller profiles with respect to similarity of a buyer profile. Result of such a computation is a value between 0 and 1, where 1 signifies 100% match and 0 signifies no match, 0.5 signifies 50-50 match, and so on. A preview of search results is shown in Figure 5.7. Key

information provided in this view includes the seller name, percentage relevance of seller profile in relation to the buyer profile, check box for potential partner selection.



Offer Name	%Relevance	Score	Potential Partner
Seller3	85%	77	<input type="checkbox"/>
Seller1	77%	70	<input type="checkbox"/>
Seller4	66.67%	60	<input type="checkbox"/>
Seller2	44.45%	40	<input type="checkbox"/>
Seller5	42.23%	38	<input type="checkbox"/>

Save Partners

Figure 5.7: Search Results showing the ranked list of matching seller profiles to a given buyer profile.

5.5 Discussion

The process to establish a partnership is implemented and tested based on 1 buyer profile and 5 seller profiles. Buyer Profile in Figure 5.6, note that the feature **Unique Competency** has knockout attribute selected. This means, if any of the sellers do not have the feature **Unique Competency** in their seller profile or do not have the value “Automotive Manufacturing” for Unique Competency will be simply discarded. The sourcing attribute and knockout

attribute works exactly opposite of each other; one (knockout) is very strict whereas other (sourcing) is very lenient. Also, buyer1 has higher weight for following features **Cash Flow**, **Human Resources**, and **Manufacturing Skills** whereas the follower features has lower weight **Currency**, **Cultural Alignment**, and **Willingness to Share Expertise**. Higher weight for features suggests their importance and lower weight suggests that they are less important.

In this case study, all the seller profiles have the value “Automotive Manufacturing” for feature **Unique Competency** in their profile, so none of them is knocked-out. The seller with the highest score is regarded as the best performing seller and the rest can be ranked accordingly. The results, from case study, indicates that the top two sellers are seller3 and seller1 - their respective relevance percentages are 85% and 77%. We believe these sellers receive more business than any other seller, however, empirical studies have revealed that relevance score less than 50% reflects seller whose priorities do not align with buyer’s requirements. Semantic Similarity measure shows that Seller 4 is relatively better than Seller 2. For this work, we can regard 50% as cut off value. Note that, a buyer is choose to free the cut-off point, it can be a percentage relevance or top 5 or top10. He can then negotiate with the seller and further align their respective ambitions. The main advantages can be described as follows

1. The proposed methodology for partnership establishment allows selecting sellers in a global environment thus enables sellers to expand themselves

globally. The system provides an access point for buyers to source partners in globally disperse developed and developing countries. Therefore, it allows buyers to embark into emerging markets such as China, India and reduce their manufacturing costs, resources, and gain expertise.

2. Generating, storing, manipulating, and distributing information is central to a successful partner establishment process. The challenge of making relevant information available in distributed partnership establishment is addressed by Partnership Ontology. The problem of synonymy and polysemy is taken care of by the UNSPSC ontology. Ontology in this case allows machine readable representation of buyer profiles and seller profiles. Some of the other advantages that come with the use of ontologies is that they are easy to update, can easily borrow concepts and properties from other ontologies and expand themselves, can be merged together with other ontologies, etc.

5.6 Conclusions

Most of the research work in partnership establishment rank sellers, given buyer requirements. They use various mathematical models like AI, Neural Network, DES, Analytic Hierarchy Process (AHP), and Quality Function Deployment (QFD). To the best of our knowledge, no work exists that have addressed the integration problem in partnership establishment process. In this work, we capitalise on ontologies to provide a machine readable representation of buyer

and seller profiles, propose a semantic similarity measure to rank seller profiles for a given buyer profile. We also implemented a web service that automates the whole process from representation of profiles to final ranking of seller profiles. It is evident from the results, analysis and the discussion outlined in the previous sections that the methodology presented in this chapter is a feasible, useful and practical for ranking buyer-seller in a globalized situation. The proposed methodology is unique in the sense that ontologies are employed and vector space model is used so as to provide a solid systematic approach which is also mathematically proven. The major innovation of the proposed methodology is that the UNSPSC ontology provides a unique code for **manufacturing skills** that helps in disambiguation of any product or services. Classifying products and services with a common coding scheme facilitates commerce between buyers and sellers and is becoming mandatory in the new era of electronic commerce. There are some delicate issues like privacy, cultural, intellectual property rights, etc that needs to be addressed in this research. As a future work, this work can be extended for the ownership type partnerships or joint ventures etc. To extend this work, such that, multiple SMEs or partners be selected for a given job and how to distribute jobs among them is an interesting research problem

.

6

Conclusion

Now this is not the end. It is not even the beginning of the end. But it is, perhaps, the end of the beginning. - Winston Churchill

In this dissertation, I worked on different manifestations of user profile for different domains. In the domain of personalized search, a user profile is manifested as User Interest Profile (*UIP*) and Clustered User Interest Profile (*CUIP*). I proposed three novel methods that exploited user search history and social bookmarking services for building a User Interest Profile(*UIP*) and Clustered User Interest Profile (*CUIP*) that consists of term clusters of user interests. The first method for personalized search is termed as 'Exclusively Yours'. It builds a *UIP* from the anchor text of hub pages of the user clicked Web documents. We also proposed a method to calculate the term-weights that originates from multiple documents and are accumulated in the *UIP*. After the construction of a *UIP*, we propose a query expansion method that relies on information distance and discounts the terms that have not been updated for a time dura-

tion, thus, logically segregating a *UIP* into two parts. The proposed method is compared against non-folksonomy based personalized search methods and non-personalized search using the Precision, Discounted Cumulative Gain (DCG), and Average Rank (AR) evaluation metrics. It has demonstrated improved search quality against its comparators. The results were satisfactory but it has its own limitations. We found that a *UIP* constructed from anchor text also has some unintentional noise embedded into it.

The second method, to construct a *UIP* and *CUIP*, is based on the Singular Value Decomposition (SVD) to compute a tag-tag similarity matrix and use the Hierarchical Agglomerative Clustering (HAC) on the matrix to generate a cluster structure, *svdCUIP*. The third method is an extension of the first method, called modified Singular Value Decomposition (modSVD), that aims to group related tags based on their second-order co-occurrence similarity. This method is based on the assumption that related tags are often expressed together by similar sets of tags. These semantically related tags are bound to co-occur with similar neighbours. The objective of the modSVD is to discover and group these semantically related tags into clusters to generate a *modSvdCUIP*, each cluster of which identifies a unified topic. For these two methods, we proposed an automatic evaluation method that does not require user involvement to enumerate the relevancy of search results. We found out it to be an effective method to compare personalized search methods.

To evaluate the effectiveness of the proposed approaches, we compared them with the baseline search and the three other methods that use folksonomy for

constructing *UIP* and Resource Profile (*RP*): *tfUIP* (Noll and Meinel, 2007), *tfIdfUIP* (Xu, Bao, Fei, Su, and Yu, 2008), *tfIdfCUIP*. Our methods are more realistic as they make no assumption about the tagging activity of the user, and can be easily put to practice for any user who uses a search engine for his/her daily search needs. In our evaluations, we found that the improvement in the ranking scores of the target URLs for the *modSvdCUIP* based personalized search were better than all the other methods; the *modSvdCUIP* approach showed improvement of 71.6%, 27.8%, 12%, 6.6%, and 8.1% over the baseline (Lucene Search), *tfIdfUIP*, *tfUIP*, *tfIdfCUIP*, and *svdCUIP* approaches, respectively.

All three proposed methods are non-invasive. In other words, they make no attempt to collect user personal information. The only objective is to mine user interests and find relationship between them. Each cluster, in the cluster structure *CUIP*, identifies a distinct topic, and the application of *CUIP* aids in disambiguating the context of use query, which is particularly needed for vague queries. It is also very effective in disambiguating the synonymy and polysemy terms.

In the domain of Partnership Match, a user profile is manifested as a buyer profile or seller profile which is drawn from a controlled vocabulary. The controlled vocabulary in this case is an ontology. I also proposed an ontology, termed as partnership ontology, which contains the concepts and relationship between them. A semantic similarity measure based on Vector Space Model is proposed to score and rank seller profiles for a given buyer profile. To the

best of our knowledge, no work exists that have addressed the integration problem in partnership establishment process. The partnership ontology provides a machine readable representation of buyer and seller profiles. The proposed methodology is unique in the sense that ontologies are employed and vector space model is used so as to provide a solid systematic approach which is also mathematically proven.

6.1 Future Work

Last, but not least, several issues need to be targeted to improve the personalized search and partnership match. In the next two subsection, I talk about the future wor in the domain of personalized search and the last section is about partnership match.

6.1.1 Degree of Personalization

Experiment results in personalized search suggest that not all queries need personalization. One task that remains outstanding is how to determine which query needs personalization and which does not. This task can be, to some extent, tackled by classifying the nature of the queries(Broder, 2002): navigational, Informational queries, transactional queries. We also observed in our experiments that navigational queries do not need disambiguation. For instance, the topmost result for the query "jigsaw puzzle" is <http://www.zigzone.com>, which is the best possible match; the query "jigsaw puzzle" does not require

any disambiguation. However, information queries, for instance "puzzle game", that cover a broad range of topics can be benefited by personalization; part of the reason is user's inability to represent his information needs in 2 or 3 words (Amanda, Dietmar, Major, and Tefko, 2001), the average length of user's query. It is easy to determine the type of query by using statistical methods (Rose and Levinson, 2004) or using machine learning approaches (Beitzel et al., 2005). It is the need of the hour that a personalized search web service should automatically diagnose the nature of input query and decide if it needs to be disambiguated or not.

6.1.2 Filter Bubble

A contrarian view to personalized search is "**Filter Bubble**". According to Wikipedia¹, a filter bubble is a result state in which a search algorithm selectively guesses what information a user would like to be interested in based on interests of the user which are largely derived from the user past click behavior (search history), twitter posts, Web pages visited. Some of the examples are Google's Personalized Search, Facebook recommendations, twitter news recommendation, and so on. This term was coined by internet activist Eli Pariser in his book (Pariser, 2011) that states, "users get less exposure to conflicting viewpoints and are isolated intellectually in their own information bubble". In other words, the information bubble subdues serendipity which closes us off to new ideas, subjects, and important information. In my future work, I would like

¹http://en.wikipedia.org/wiki/Filter_bubble

to study the effect and magnitude of information bubble on personalization so that a quantifiable measure can be developed to calculate the effect. This in turn might also provide directions in drawing a balance between personalization and information bubble.

I will also look into more advanced methods such as probabilistic LSI and Latent Dirichlet Allocation(LDA) for discovering and building a more efficient *CUIP*.

6.1.3 IPR issues in Partnership Match

One of the issues that needs to be addressed is intellectual property rights (IPR), it needs to be protected during the partnership establishment process. Several sophisticated methods for information exchange via the Internet are being developed, however, end users are reluctant to share their information on-line. For the future research, I would like to focus on how to embed trust in user profiles (buyer profile or seller profile) in the partnership match, and how to control access to information during partnership establishment.

Bibliography

- F. Abel, Q. Gao, G.-J. Houben, and K. Tao. Analyzing user modeling on twitter for personalized news recommendations. In *Proceedings of the 19th international conference on User modeling, adaption, and personalization, UMAP'11*, pages 1–12, Berlin, Heidelberg, 2011. Springer-Verlag. ISBN 978-3-642-22361-7. 5, 45
- D. Achlioptas and F. McSherry. Fast computation of low rank matrix approximations. In *Proceedings of the thirty-third annual ACM symposium on Theory of computing*, pages 611–618. ACM, 2001. 41
- E. Agichtein, E. Brill, and S. Dumais. Improving web search ranking by incorporating user behavior information. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval, SIGIR '06*, pages 19–26, New York, NY, USA, 2006. ACM. ISBN 1-59593-369-7. 43, 48, 87

BIBLIOGRAPHY

- F. Alan, K. Ravi, and V. Santosh. Fast monte-carlo algorithms for finding low-rank approximations. *Journal of the ACM*, 51(6):1025-1041, 2004. 41
- S. Amanda, W. Dietmar, B. J. J. Major, and S. Tefko. Searching the web: The public and their queries. *Journal of the American Society for Information Science and Technology*, 52:226-234, 2001. 168
- S. Andriy, G. Jonathan, M. Bamshad, and D. B. Robin. Personalized recommendation in social tagging systems using hierarchical clustering. In *RecSys*, pages 259-266, 2008. 45, 46, 47, 50, 86, 101, 129, 130
- C. Basnet and J. M. Leung. Inventory lot-sizing with supplier selection. *Computers & Operations Research*, 32(1):1-14, 2005. 52, 54
- O. Bayazit. Use of analytic network process in vendor selection decisions. *Benchmarking: An International Journal*, 13(5):566-579, 2006. 53, 144
- I. H. Beaumont. User modelling in the interactive anatomy tutoring system anatom-tutor. *User Modeling and User-Adapted Interaction*, 4(1):21-45, 1994. 4
- S. M. Beitzel, E. C. Jensen, O. Frieder, D. Grossman, D. D. Lewis, A. Chowdhury, and A. Kolcz. Automatic web query classification using labeled and unlabeled training data. In *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '05, pages 581-582, New York, NY, USA, 2005. ACM. ISBN 1-59593-034-5. 168

BIBLIOGRAPHY

- M. W. Berry, M. Browne, A. N. Langville, V. P. Pauca, and R. J. Plemmons. Algorithms and applications for approximate nonnegative matrix factorization. *Computational Statistics & Data Analysis*, 52(1):155–173, 2007. 42
- L. Bing. *Web Data Mining: Exploring Hyperlinks, Contents, and Usage Data (Data-Centric Systems and Applications)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006. ISBN 3540378812. 104, 106
- D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *the Journal of machine Learning research*, 3:993–1022, 2003. 42
- M. R. Bouadjenek, H. Hacid, and M. Bouzeghoub. Laicos: an open source platform for personalized social web search. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1446–1449. ACM, 2013a. 50
- M. R. Bouadjenek, H. Hacid, M. Bouzeghoub, and A. Vakali. Using social annotations to enhance document representation for personalized search. In *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval*, pages 1049–1052. ACM, 2013b. 50
- C. Boyle and A. O. Encarnacion. Metadoc: an adaptive hypertext reading system. In *Adaptive Hypertext and Hypermedia*, pages 71–89. Springer, 1998. 4

BIBLIOGRAPHY

- T. J. Brailsford, C. D. Stewart, M. R. Zakaria, and A. Moore. Autonavagation, links and narrative in an adaptive web-based integrated learning environment. 2002. 4
- G. Brajnik, G. Guida, and C. Tasso. User modeling in intelligent information retrieval. *Information Processing & Management*, 23(4):305–320, 1987. 6
- S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. *Comput. Netw. ISDN Syst.*, 30:107–117, April 1998. ISSN 0169-7552. 43, 57
- A. Broder. A taxonomy of web search. *SIGIR Forum*, 36:3–10, September 2002. ISSN 0163-5840. 131, 167
- P. Brusilovsky and D. W. Cooper. Domain, task, and user models for an adaptive hypermedia performance support system. In *Proceedings of the 7th international conference on Intelligent user interfaces*, pages 23–30. ACM, 2002. 6
- P. Brusilovsky, A. Kobsa, and W. Nejdl. *The adaptive web: methods and strategies of web personalization*, volume 4321. Springer, 2007. 1
- P. L. Brusilovsky. A framework for intelligent knowledge sequencing and task sequencing. In *Intelligent tutoring systems*, pages 499–506. Springer, 1992. 6
- W. Buntine. Variational extensions to em and multinomial pca. In *Machine Learning: ECML 2002*, pages 23–34. Springer, 2002. 42

BIBLIOGRAPHY

- W. Buntine and A. Jakulin. Discrete component analysis. In *Subspace, Latent Structure and Feature Selection*, pages 1–33. Springer, 2006. 42
- A. Cakravastia and K. Takahashi*. Integrated model for supplier selection and negotiation in a make-to-order environment. *International Journal of Production Research*, 42(21):4457–4474, 2004. 52, 54
- J. M. Carroll and M. B. Rosson. Interfacing thought: cognitive aspects of human-computer interaction. chapter Paradox of the active user, pages 80–111. MIT Press, Cambridge, MA, USA, 1987. ISBN 0-262-03125-6. 44
- S. Chakrabarti, B. Dom, D. Gibson, S. R. Kumar, P. Raghavan, S. Rajagopalan, and A. Tomkins. Experiments in topic distillation. In *ACM SIGIR workshop on Hypertext Information Retrieval on the Web*. Melbourne, Australia, 1998a. 57
- S. Chakrabarti, B. Dom, P. Raghavan, S. Rajagopalan, D. Gibson, and J. Kleinberg. Automatic resource compilation by analyzing hyperlink structure and associated text. *Computer Networks and ISDN Systems*, 30(1):65–74, 1998b. 57
- C.-T. Chen, C.-T. Lin, and S.-F. Huang. A fuzzy approach for supplier evaluation and selection in supply chain management. *International Journal of Production Economics*, 102(2):289–301, 2006. 52, 53
- S.-H. Chen, H.-T. Lee, and Y.-F. Wu. Applying anp approach to partner

BIBLIOGRAPHY

- selection for strategic alliance. *Management Decision*, 46(3):449–465, 2008. 52, 144
- P.-A. Chirita, C. S. Firan, and W. Nejdl. Summarizing local context to personalize global web search. In *Proceedings of the 15th ACM international conference on Information and knowledge management, CIKM '06*, pages 287–296, New York, NY, USA, 2006. ACM. ISBN 1-59593-433-2. 43, 44, 45, 48, 57, 59
- K. Choy and W. Lee. A generic supplier management tool for outsourcing manufacturing. *Supply Chain Management: An International Journal*, 8(2): 140–154, 2003. 52, 53, 144
- M. D. V. Christopher, G. Shlomo, and T. Andrew. Document clustering evaluation: Divergence from a random baseline. *CoRR*, abs/1208.5654, 2012. 104
- J. E. Cohen and U. G. Rothblum. Nonnegative ranks, decompositions, and factorizations of nonnegative matrices. *Linear Algebra and its Applications*, 190:149–168, 1993. 42
- Y. Crama, R. Pascual J, and A. Torres. Optimal procurement decisions in the presence of total quantity discounts and alternative product recipes. *European Journal of Operational Research*, 159(2):364–378, 2004. 53
- M. T. Dacin, M. A. Hitt, and E. Levitas. Selecting partners for successful

BIBLIOGRAPHY

- international alliances: examination of us and korean firms. *Journal of world business*, 32(1):3–16, 1997. 144
- A. S. Das, M. Datar, A. Garg, and S. Rajaram. Google news personalization: scalable online collaborative filtering. In *Proceedings of the 16th international conference on World Wide Web, WWW '07*, pages 271–280, New York, NY, USA, 2007. ACM. ISBN 978-1-59593-654-7. 44, 57
- V. David, C. Iván, and M. J. Joemon. Personalizing web search with folksonomy-based user and document profiles. In *ECIR*, pages 420–431, 2010. 12, 45, 48, 50, 125, 128, 129
- G. W. Dickson. An analysis of vendor selection systems and decisions. *Journal of purchasing*, 2(1):5–17, 1966. 53, 54, 141
- L. Ding, T. Finin, A. Joshi, R. Pan, R. S. Cost, Y. Peng, P. Reddivari, V. Doshi, and J. Sachs. Swoogle: a search and metadata engine for the semantic web. In *Proceedings of the thirteenth ACM international conference on Information and knowledge management*, pages 652–659. ACM, 2004. 83
- B. E. Dom. An information-theoretic external cluster-validity measure. In *Proceedings of the Eighteenth conference on Uncertainty in artificial intelligence, UAI'02*, pages 137–145, San Francisco, CA, USA, 2002. Morgan Kaufmann Publishers Inc. ISBN 1-55860-897-4. 104
- Z. Dou, R. Song, and J.-R. Wen. A large-scale evaluation and analysis of personalized search strategies. In *Proceedings of the 16th international con-*

BIBLIOGRAPHY

- ference on World Wide Web*, WWW '07, pages 581–590, New York, NY, USA, 2007. ACM. ISBN 978-1-59593-654-7. 48
- P. Drineas, R. Kannan, and M. W. Mahoney. Fast monte carlo algorithms for matrices ii: Computing a low-rank approximation to a matrix. *SIAM Journal on Computing*, 36(1):158–183, 2006a. 41
- P. Drineas, R. Kannan, and M. W. Mahoney. Fast monte carlo algorithms for matrices iii: Computing a compressed approximate matrix decomposition. *SIAM Journal on Computing*, 36(1):184–206, 2006b. 41
- P. Drineas, M. W. Mahoney, and S. Muthukrishnan. Subspace sampling and relative-error matrix approximation: Column-based methods. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, pages 316–326. Springer, 2006c. 41
- P. Drineas, M. W. Mahoney, and S. Muthukrishnan. Subspace sampling and relative-error matrix approximation: Column-row-based methods. In *Algorithms-ESA 2006*, pages 304–314. Springer, 2006d. 41
- P. Drineas, M. W. Mahoney, and S. Muthukrishnan. Relative-error cur matrix decompositions. *SIAM Journal on Matrix Analysis and Applications*, 30(2): 844–881, 2008. 41
- R. Dulmin and V. Mininno. Supplier selection using a multi-criteria decision aid method. *Journal of Purchasing and Supply Management*, 9(4):177–187, 2003. 52, 53

BIBLIOGRAPHY

- J. H. Dyer. *Collaborative advantage: Winning through extended enterprise supplier networks*. Oxford University Press New York, 2000. 144
- M. V. Ellen. The trec-8 question answering track report. In *In Proceedings of TREC-8*, pages 77–82, 1999. 108
- D. C. Engelbart. Augmenting Human Intellect: A Conceptual Framework. Air Force Office of Scientific Research, AFOSR-3233, www.bootstrap.org/augdocs/friedewald030402/augmentinghumanintellect/ahi62index.html, 1962. 37
- A. Eugene, B. Eric, D. Susan, and R. Robert. Learning user interaction models for predicting web search result preferences. In *SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 3–10. ACM, 2006. 43
- A. Fabian, H. Nicola, H. Eelco, and K. Daniel. Interweaving public user profiles on the web. In *UMAP*, pages 16–27. Springer, 2010. 45
- D. Fensel. *Ontologies: A silver bullet for knowledge management and electronic-commerce (2000)*. Berlin: Spring-Verlag. 143
- P. Ferragina and A. Gulli. A personalized search engine based on web-snippet hierarchical clustering. In *Special interest tracks and posters of the 14th international conference on World Wide Web, WWW '05*, pages 801–810, New York, NY, USA, 2005. ACM. ISBN 1-59593-051-5. 48, 49, 57, 59

BIBLIOGRAPHY

- T. Fred. *From Counterculture to Cyberculture: Stewart Brand, the Whole Earth Network, and the Rise of Digital Utopianism*. The University of Chicago Press, Chicago, Ill., 2006. ISBN 0-226-81741-5. 37
- S. Gauch, J. Chaffee, and A. Pretschner. Ontology based personalized search and browsing. *Web Intelli. and Agent Sys.*, 1:219-234, December 2003. ISSN 1570-1263. 45, 48, 57
- H. G. Gene and F. V. L. Charles. *Matrix Computations*. 1996. 40, 41
- R. E. Giachetti. A framework to review the information integration of the enterprise. *International Journal of Production Research*, 42(6):1147-1166, 2004. 135
- J. C. Gower and G. Ross. Minimum spanning trees and single linkage cluster analysis. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 18(1):54-64, 1969. 99
- C. Hans. Supporting partner identification for virtual organisations in manufacturing. *Journal of Manufacturing Technology Management*, 19(4):497-513, 2008. 144
- H. M. Hassan. Clustering web images using association rules, interestingness measures, and hypergraph partitions. In *In: ICWE 06: Proceedings of the 6th international conference on Web engineering*, pages 48-55. ACM Press, 2006. 104

BIBLIOGRAPHY

- T. H. Haveliwala. Topic-sensitive pagerank. In *Proceedings of the 11th international conference on World Wide Web*, pages 517–526. ACM, 2002. 57
- M. Heckner, M. Heilemann, and C. Wolff. Personal information management vs. resource sharing: Towards a model of information behavior in social tagging systems. In *ICWSM, 2009*. 39
- M. Hepp. Goodrelations: An ontology for describing products and services offers on the web. In *Knowledge Engineering: Practice and Patterns*, pages 329–346. Springer, 2008. 140
- T. Hofmann. Probabilistic latent semantic indexing. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 50–57. ACM, 1999a. 42
- T. Hofmann. Probabilistic latent semantic analysis. In *Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence*, pages 289–296. Morgan Kaufmann Publishers Inc., 1999b. 42
- T. Hofmann. Unsupervised learning by probabilistic latent semantic analysis. *Machine learning*, 42(1-2):177–196, 2001. 42
- A. Hotho, R. Jäschke, C. Schmitz, and G. Stumme. Emergent semantics in bibsonomy. *GI Jahrestagung (2)*, 94:305–312, 2006. 38
- A. Ioannis, A. Konstantinos, and M. J. Joemon. A comparison of general vs personalised affective models for the prediction of topical relevance. In *SIGIR*, pages 371–378, 2010. 10

BIBLIOGRAPHY

- P. K. Jagersma and D. M. van Gorp. Still searching for the pot of gold: doing business in todays china. *Journal of Business Strategy*, 24(5):27–35, 2003. 144
- K. Järvelin and J. Kekäläinen. Cumulated gain-based evaluation of ir techniques. *ACM Trans. Inf. Syst.*, 20:422–446, October 2002. ISSN 1046-8188. 73, 74
- H. Jiawei and K. Micheline. *Data Mining: Concepts and techniques*. 2001. 40
- N. J. Kaplan and J. Hurd. Realizing the promise of partnerships. *Journal of Business Strategy*, 23(3):38–42, 2002. 144
- L. Kaufman and P. J. Rousseeuw. *Finding Groups in Data: An Introduction to Cluster Analysis*. John Wiley, 1990. ISBN 1-58133-109-7. 106
- J. Kay and R. Kummerfeld. An individualised course for the c programming language. In *Proceedings of Second International WWW Conference*, pages 17–20, 1994. 7
- D. Kelly and J. Teevan. Implicit feedback for inferring user preference: a bibliography. *SIGIR Forum*, 37:18–28, September 2003. ISSN 0163-5840. 43
- S. Koshman, A. Spink, and B. J. Jansen. Web searching on the vivisimo search engine. *J. Am. Soc. Inf. Sci. Technol.*, 57:1875–1887, December 2006. ISSN 1532-2882. 48, 57

BIBLIOGRAPHY

- R. Kraft and J. Zien. Mining anchor text for query refinement. In *Proceedings of the 13th international conference on World Wide Web*, pages 666–674. ACM, 2004. 62
- A. Krüger, J. Baus, D. Heckmann, M. Kruppa, and R. Wasinger. Adaptive mobile guides. In *The adaptive web*, pages 521–549. Springer, 2007. 8
- H. Kumar and S. Kang. Another face of search engine: Web search api’s. In *Proceedings of the 21st international conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems: New Frontiers in Applied Artificial Intelligence, IEA/AIE ’08*, pages 311–320, Berlin, Heidelberg, 2008. Springer-Verlag. ISBN 978-3-540-69045-0. 48
- H. Kumar and H.-G. Kim. Semantically enriched user interest profile built from users’ tweets. In *ICADL*, pages 333–337, 2012. 39, 45
- H. Kumar and H.-G. K. Kim. Using folksonomies for building user interest profile. In *UMAP*, pages 438–441, 2011. 39, 45
- D. D. Lee and H. S. Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–791, 1999. 42
- J. S. Lee, N. Lee, C. H. Noh, and Y. Han. Ontology-based job scheduling using mobile agent technology in grid computing. *Information: An International Interdisciplinary Journal*, 13(5):1639–1651, 2010. 54
- K. Lee, H. Kim, H. Shin, and H.-J. Kim. Tag sense disambiguation for clarifying the vocabulary of social tags. In *Computational Science and Engineering*,

BIBLIOGRAPHY

2009. *CSE'09. International Conference on*, volume 4, pages 729–734. IEEE, 2009. 39
- F. Lemke, K. Goffin, and M. Szwejczeński. Investigating the meaning of supplier-manufacturer partnerships: an exploratory study. *International Journal of Physical Distribution & Logistics Management*, 33(1):12–35, 2003. 144
- L. Li, B. Wu, and Y. Yang. An ontology-oriented approach for virtual enterprises. In *Advanced Web Technologies and Applications*, pages 834–843. Springer, 2004a. 52, 54
- M. Li, X. Chen, X. Li, B. Ma, and P. M. Vitányi. The similarity metric. *Information Theory, IEEE Transactions on*, 50(12):3250–3264, 2004b. 65, 67
- Y. Li, B. Huang, W. Liu, H. Gou, and C. Wu. Ontology based decision support system for partner selection of virtual enterprises. In *Systems, Man, and Cybernetics, 2001 IEEE International Conference on*, volume 3, pages 2041–2045. IEEE, 2001. 52, 54
- J. Lin, X. Xu, and D. Xu. Strategic supplier selection: A domain driven data mining methodology. *Information*, 13(4):1449–1465, 2010. 52
- F.-H. F. Liu and H. L. Hai. The voting analytic hierarchy process method for selecting supplier. *International Journal of Production Economics*, 97(3): 308–317, 2005. 52, 144

BIBLIOGRAPHY

- J. López, E. Millán, J. Pérez-de-la Cruz, and F. Triguero. Ilesa: a web-based intelligent learning environment for the simplex algorithm. In *Proc. of CALISCE*, volume 98, pages 399–406, 1998. 4
- Z. Ma, G. Pant, and O. R. L. Sheng. Interest-based personalized search. *ACM Trans. Inf. Syst.*, 25, February 2007. ISSN 1046-8188. 43
- B. Maheshwari, V. Kumar, and U. Kumar. Optimizing success in supply chain partnerships. *Journal of Enterprise Information Management*, 19(3):277–291, 2006. 144
- C. D. Manning, P. Raghavan, and H. Schtze. *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA, 2008. ISBN 0521865719, 9780521865715. 73
- D. McArthur, C. Stasz, J. Hotta, O. Peter, and C. Burdorf. Skill-oriented task sequencing in an intelligent tutor for basic algebra. *Instructional Science*, 17(4):281–307, 1988. 6
- O. A. McBryan. Genvl and www: Tools for taming the web. In *Proceedings of the first international world wide web conference*, volume 341. Geneva, 1994. 62
- G. I. McCalla, R. B. Bunt, and J. J. Harms. The design of the scent automated advisor. *Computational Intelligence*, 2(1):76–92, 1986. 6
- H. Min. International supplier selection:: A multi-attribute utility approach.

BIBLIOGRAPHY

- International Journal of Physical Distribution & Logistics Management*, 24 (5):24–33, 1994. 52, 53
- M. G. Noll and C. Meinel. Web search personalization via social bookmarking and tagging. In *Proceedings of the 6th international The semantic web and 2nd Asian conference on Asian semantic web conference, ISWC'07/ASWC'07*, pages 367–380, Berlin, Heidelberg, 2007. Springer-Verlag. ISBN 3-540-76297-3, 978-3-540-76297-3. xii, 12, 13, 45, 46, 47, 48, 49, 85, 128, 166
- Y. Okazaki, K. Watanabe, and H. Kondo. An implementation of the www based its for guiding differential calculations. In *Proc. of Workshop " Intelligent Educational Systems on the World Wide Web" at 8th World Conference on Artificial Intelligence in Education*, pages 18–25, 1997. 4
- P. Paatero and U. Tapper. Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values. *Environmetrics*, 5(2):111–126, 1994. 42
- C. H. Papadimitriou, H. Tamaki, P. Raghavan, and S. Vempala. Latent semantic indexing: A probabilistic analysis. In *Proceedings of the seventeenth ACM SIGACT-SIGMOD-SIGART symposium on Principles of database systems*, pages 159–168. ACM, 1998. 42
- E. Pariser. *The Filter Bubble: What the Internet Is Hiding from You*. Penguin Press, New York, NY, USA, 2011. ISBN 978-1-59420-300-8. 168

BIBLIOGRAPHY

- G.-P. A. Pérez, A. Zubiaga, V. Fresno, and R. Martínez. Reorganizing clouds: A study on tag clustering and evaluation. *Expert Syst. Appl.*, 39(10):9483–9493, Aug. 2012. ISSN 0957-4174. 104
- S. A. Petersen and M. Divitini. Using agents to support the selection of virtual enterprise teams. *AOIS@ AAMAS*, 59, 2002. 52, 54
- X. H. Pham and J. J. Jung. Exploiting semantic template for message summarization for mobile devices. *Information: An International Interdisciplinary Journal*, 13(4):1467–1474, 2010. 54
- T. A. Phelps and R. Wilensky. Robust hyperlinks cost just five words each. 2000. 68
- A. B. Pidduck. Issues in supplier partner selection. *Journal of Enterprise Information Management*, 19(3):262–276, 2006. 144
- J. Pitkow, H. Schütze, T. Cass, R. Cooley, D. Turnbull, A. Edmonds, E. Adar, and T. Breuel. Personalized search. *Commun. ACM*, 45(9):50–55, Sept. 2002. ISSN 0001-0782. 48
- S. Rennick-Egglestone, A. Whitbrook, C. Leygue, J. Greensmith, B. Walker, S. Benford, H. Schnädelbach, S. Reeves, J. Marshall, D. Kirk, et al. Personalizing the theme park: psychometric profiling and physiological monitoring. In *User Modeling, Adaption and Personalization*, pages 281–292. Springer, 2011. 5

BIBLIOGRAPHY

- R. Riding and S. Rayner. *Cognitive styles and learning strategies: Understanding style differences in learning and behaviour*. D. Fulton Publishers, 1998. 7
- D. E. Rose and D. Levinson. Understanding user goals in web search. In *Proceedings of the 13th international conference on World Wide Web, WWW '04*, pages 13–19, New York, NY, USA, 2004. ACM. ISBN 1-58113-844-X. 168
- G. Rossi, D. Schwabe, and R. Guimarães. Designing personalized web applications. In *Proceedings of the 10th international conference on World Wide Web*, pages 275–284. ACM, 2001. 5
- P. J. Rousseeuw. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20(1):53–65, Nov. 1987. ISSN 03770427. 105
- G. Salton. The smart retrieval system experiments in automatic document processing. 1971. 56
- G. Salton, A. Wong, and C.-S. Yang. A vector space model for automatic indexing. *Commun. ACM*, 18(11):613–620, 1975. 102, 155
- J. Schmitz and K. Platts. Supplier logistics performance measurement: indications from a study in the automotive industry. *International Journal of Production Economics*, 89(2):231–243, 2004. 53

BIBLIOGRAPHY

- P. Schmitz. Inducing ontology from flickr tags. In *Collaborative Web Tagging Workshop at WWW2006, Edinburgh, Scotland*, volume 50, 2006. 39
- D. Scott, T. D. Susan, W. F. George, K. L. Thomas, and H. Richard. Indexing by latent semantic analysis. *JOURNAL OF THE AMERICAN SOCIETY FOR INFORMATION SCIENCE*, 41(6):391–407, 1990. 91, 98
- S. Shekhar. Benchmarking knowledge gaps through role simulations for assessing outsourcing viability. *Benchmarking: An International Journal*, 15(3): 225–241, 2008. 144
- X. Shen, B. Tan, and C. Zhai. Context-sensitive information retrieval using implicit feedback. In *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval, SIGIR '05*, pages 43–50, New York, NY, USA, 2005. ACM. ISBN 1-59593-034-5. 43, 44
- B. Sigurbjörnsson and Z. R. Van. Flickr tag recommendation based on collective knowledge. In *Proceedings of the 17th international conference on World Wide Web*, pages 327–336. ACM, 2008. 39
- A. P. Singh and G. J. Gordon. A unified view of matrix factorization models. In *Machine Learning and Knowledge Discovery in Databases*, pages 358–373. Springer, 2008. 42
- M. Speretta and S. Gauch. Personalized search based on user search histories. In *Proceedings of the 2005 IEEE/WIC/ACM International Conference on*

BIBLIOGRAPHY

- Web Intelligence*, WI '05, pages 622–628, Washington, DC, USA, 2005a. IEEE Computer Society. ISBN 0-7695-2415-X. 45, 48
- M. Speretta and S. 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, 2005b. 57
- Z. Sui and Q. Zhao. To extract ontology attribute value automatically based on www. *Information: An International Interdisciplinary Journal*, 12(2), 2009. 54
- J.-T. Sun, H.-J. Zeng, H. Liu, Y. Lu, and Z. Chen. Cubesvd: a novel approach to personalized web search. In *Proceedings of the 14th international conference on World Wide Web*, pages 382–390. ACM, 2005. 57
- Q. Sun, J. Ji, and W. Xu. A new approach for vendor evaluation and selection based on maximizing deviation multiple attribute decision. *Information: An International Interdisciplinary Journal*, 12(1):13–19, 2009. 52, 53
- P.-N. Tan, M. Steinbach, and V. Kumar. *Introduction to Data Mining, (First Edition)*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2005. ISBN 0321321367. 113
- F. Tarpin-Bernard and H. Habieb-Mammar. Modeling elementary cognitive abilities for adaptive hypermedia presentation. *User Modeling and User-Adapted Interaction*, 15(5):459–495, 2005. 8

BIBLIOGRAPHY

- J. Teevan, S. Dumais, and E. Horvitz. Personalizing search via automated analysis of interests and activities. In *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '05, 2005. ISBN 1-59593-034-5. 43, 44, 57, 59
- J. Teevan, S. T. Dumais, and E. Horvitz. Characterizing the value of personalizing search. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '07, pages 757–758, New York, NY, USA, 2007. ACM. ISBN 978-1-59593-597-7. 43, 57
- O. Tim. What is web 2.0: Design patterns and business models for the next generation of software. 2005. doi: <http://dx.doi.org/10.1016/j.websem.2007.11.011>. URL <http://www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html>. 36
- G. Tom. Collective knowledge systems: Where the social web meets the semantic web. *Web Semantics: Science, Services and Agents on the World Wide Web*, 6(1):4 – 13, 2008. ISSN 1570-8268. doi: <http://dx.doi.org/10.1016/j.websem.2007.11.011>. URL <http://www.sciencedirect.com/science/article/pii/S1570826807000583>. 35
- V. Tsiriga and M. Virvou. Modelling the student to individualise tutoring in a web-based icall. *International Journal of Continuing Engineering Education and Life Long Learning*, 13(3):350–365, 2003. 4

BIBLIOGRAPHY

- V. Uren, P. Cimiano, J. Iria, S. Handschuh, M. Vargas-Vera, E. Motta, and F. Ciravegna. Semantic annotation for knowledge management: Requirements and a survey of the state of the art. *Web Semantics: science, services and agents on the World Wide Web*, 4(1):14–28, 2006. 83
- D. Vallet and P. Castells. Personalized diversification of search results. In *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '12, pages 841–850, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1472-5. 45
- E. Van Couvering. The history of the internet search engine: Navigational media and the traffic commodity. In *Web Search*, pages 177–206. Springer, 2008. 56
- W. T. Vander. Explaining and showing broad and narrow folksonomies. <http://www.vanderwal.net/random/entrysel.php?blog=1635>, 2005. 39
- W. T. Vander. Folksonomy. <http://www.vanderwal.net/essays/051130/folksonomy.pdf>, Feb 2007. 39
- J. Vassileva. An architecture and methodology for creating a domain-independent, plan-based intelligent tutoring system. *Programmed Learning and Educational Technology*, 27(4):386–397, 1990. 6
- J. Vassileva. A task-centered approach for user modeling in a hypermedia office documentation system. *User modeling and user-adapted interaction*, 6(2-3): 185–223, 1996. 7

BIBLIOGRAPHY

- L. Wang and P. Kess. Partnering motives and partner selection: Case studies of finnish distributor relationships in china. *International Journal of Physical Distribution & Logistics Management*, 36(6):466–478, 2006. 144
- Q. Wang and H. Jin. Exploring online social activities for adaptive search personalization. In *Proceedings of the 19th ACM international conference on Information and knowledge management*, CIKM, pages 999–1008, New York, NY, USA, 2010. ACM. ISBN 978-1-4503-0099-5. 18, 48
- C. A. Weber, J. R. Current, and W. Benton. Vendor selection criteria and methods. *European journal of operational research*, 50(1):2–18, 1991. 54, 141
- S. Xu, S. Bao, B. Fei, Z. Su, and Y. Yu. Exploring folksonomy for personalized search. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '08, 2008. ISBN 978-1-60558-164-4. 12, 45, 46, 47, 49, 50, 86, 125, 128, 166

Appendices

.1 Pairs of Query and target URL

List of self-evident query and target URL

Table 1: List of Self-evident query and target URL pairs

Query	Target URL	Query	Target URL
Puzzle	zigzone.com	Math	mathlesson.com
Medicine	jmir.org	Estoer	en.wikipedia.org/ wiki/George_Gurdjieff
Hostel	en.wikibooks.org/ wiki/LaTeX/Tables	Radio	planetradiocity.com/ internetradio/index.php
amazon	amazon.com	bollywood	bollywoodhungama.com/ trade/releasedates/ index.html
Basketball	nba.com	Pbs	www.pbs.org
Islam	islamtoday.com	Boardgame	boardgamers.org
Columbia	columbia.edu	Redcross	redcross.org
Imdb	imdb.com	Thinkquest	library.thinkquest.org
Overstock	overstock.com	Gap	gap.com
Walmart	walmart.com	Ebay	cgi.ebay.com
Wikipedia	en.wikipedia.org	Citibank	citibank.com
Kraft	kraftfoods.com	Mapquest	mapquest.com
Dictionary	dictionary.com	Costco	costco.com

Continued on next page

.1 Pairs of Query and target URL

Table 1 – *Continued from previous page*

Query	Target URL	Query	Target URL
Fbi	fbi.gov	Starbucks	starbucks.com
Mtv	mtv.com	Cisco	cisco.com
Marriott	marriott.com	Weather	weather.com
Hasbro	hasbro.com	Metlife	metlife.com
Bbc	bbc.co.uk	Playboy	playboy.com
Businessweek	businessweek.com	Washingtonpost	washingtonpost.com
Whitehouse	whitehouse.gov	Time	timeanddate.com
Carter	carters.com	Skype	skype.com
Microsoft	microsoft.com	Flickr	flickr.com
Oldnavy	oldnavy.com	Patent	freepatentsonline.com
Sports	qcbaseball.com	Princeton	princeton.edu
e-health	electronic- health.org/	jigsaw puzzle	jigzone.com

List of vague query and target URL

Table 2: List of vague query and target URL pairs

Query	Target URL	Query	Target URL
Magazine	automobilemagazine.com	Planet	solarspace.co.uk

Continued on next page

.1 Pairs of Query and target URL

Table 2 – *Continued from previous page*

Query	Target URL	Query	Target URL
Auction	ragoarts.com	Worksheet	abcteach.com
Latex	betweentheshets.co.uk	Business	alibaba.com
History	onwar.com	latex	en.wikibooks.org/ wiki/LaTeX/Mathematics
Telephone	skype.com	Keynote	apple.com/ iwork/keynote/
Apple	kronenberg.org	Electronics	radioshack.com
divorce	divorcenet.com	Travel	chowbaby.com
Legal	womenslaw.com	Manufacture	tradekey.com
Realtor	foxtons.com	Food	chinesefood.about.com
Quiz	iqtest.com	Queen	queenszoo.com
Price comparison	calibex.com	Gold	Taxfreegold.co.uk
History	bible-history.com	Music	traditionalmusic.com
Entertainment	playboy.com	Database	freepatentsonline.org
Religion	cyberhymnal.org	Bible	studylight.org
Sports	qcbaseball.com	Newspaper	alligator.org
Religion	tenets.zoroastrianism.com	Stories	skywriting.net
Music	hymnal.net	Philosophy	vbm-torah.org
Automobile	kbb.com	Pond	ponds.com
Worship	Textweek.com	Health	holisticjunctino.com

Continued on next page

.2 Examples of Expanded Queries

Table 2 – *Continued from previous page*

Query	Target URL	Query	Target URL
Assist	Natri.uky.edu	Travel	ryanair.com

.2 Examples of Expanded Queries

1. The query pond was disambiguated by the cluster [beauty, products] thus pushing the `www.ponds.com` at the top of the result set.
2. The query religion is a very good example where cluster structure plays an important role. For one user who had interest in Christianity, the query religion was rightly disambiguated with the cluster [religion, Christian, church, catholic] resulting in URL `www.cyberhymnal.org` at higher rank. For another user, the same query religion was mapped to a cluster [moshiach, judaism, jewish, mysteri, mashiach, messiah] to disambiguate the context of term religion which resulted in the URL `tenets.zoronastrianism.com` promoted to the top position.
3. Another query latex was mapped to [latex, fetish, sheet, rubber, shop, house, satin, bed] pushing up the URL `www.betweenthesheets.co.uk` at the top position and lowering the rank of URLs related to Latex document markup language.

.3 An example of svdCUIP, modSvdCUIP, tfIdfCUIP

tfIdfCUIP (d=0.09)

[[ngo], [scuba, korea, dive], [editplu, softwar, regex],
[bollywood, releas, movi, hindi], [whitespace, tab, tip, format],
[data, excel, import, csv, financi, microsoft], [fm, music, radio],
[dna, genealog, genet, scienc, technolog, biologi],
[wp, wealth, wealthi, life, busi, mexico, philanthropi, person, slim,
biographi], [log, overview, classif, datamin, queri],
[video, divx, download, legenda, subtitl, film],
[free, skype, voip, telephoni, phone],
[supermercado, carrefour, casa, onlin, compra, spanish, tienda],
[comida, food, restaurant],
[mac, osx, wine, virtual, wikipedia, window, resourc, emul, linux],
[iwork, tutori, imovi, train, gwt, appl, ilif],
[lowcost, europ, vuelo, airlin, flight, lodg, travel, vacat, hotel],
[store, preppi, cheap, deal, watch, men, wear, fashion, cloth, brand,
shop, women], [financ, theater, card, bank, creditcard, cgv, samsung],
[algoritmo, poll, code, cs, binari, soa, backoff, algorithm, program,
exponentialbackoff],
[statist, decis, ahp, lean, manag, multicriteria, decisionmak, engin,

.3 An example of svdCUIP, modSvdCUIP, tflDfCUIP

projectmanag, hierarchi, analysi, process, econom, analyt],
[fourthwai, magic, spiritu, happi, learn, gurdjieff, epicurean, charact,
occult, philosophi, epicuru, esoter, osho, book],
[datetim, databas, mysql, date, creat, php, sql, exampl, function, develop],
[refer, document, latex, style, notat, packag, command, wiki, custom, tex],
[viaj, hostel, espa, airport, barcelona, spain, hostel],
[ebm, review, bmj, patient, new, cochran, socialnetwork, collabor, social,
health, commun, healthcar, medicin, medic, drug],
[openoffic], [fabul], [web2.0, semant_web, elearn, forschung, educ],
[wikibook, tabl], [float, howto, imag, figur], [firefox, extens, check],
[perform, tcpip, congest, tech, tcp, network],
[math, mathemat, verbal, teach],
[2011, confer, android:bookmark, hci, research],
[inform_scienc, inform, ci, inform-scienc, journal, li],
[chrome, webkit, tool, typographi, opensourc, typeset, browser],
[time, est, timezon, dst, convert, standard],
[matrix, librari, machin-learn, ai, java, api, algebra, machinelearn],
[load, graphic, color, comput, manual],
[entertain, kid, puzzl, interact, fun, game, jigsaw],
[informat, ehealth, internet, cfp, e-health],
[seo], [space], [paper], [export, file], [write, mactex, macosx],
[postscript, subfigur], [subscript, superscript], [shell, output],
[powerpoint, keynot, present, design],

.3 An example of svdCUIP, modSvdCUIP, tflidfCUIP

[astronomi, telrad, telescop], [cheatsheet, symbol],
[cook, restaur, vegetarian, vegan, guid],
[my.cnf, db, config, configur, backup, work, ini],
[exam, question, certif, test, scjp, mock, certification, certifica],
[babi, carter, crian, children, apparel],
[taxonomi, ux, usabl, ui, toread, ia],
[2012, lyon, public, www, www2012, via:packrati.us]]

svdCUIP(k=90, d=0.13)

[[babi, children, men, wear, fashion, cloth, brand, shop, women], [ngo],
[spiritu, happi, learn, gurdjieff, epicurean, occult, philosophi, epicuru],
[servic], [chrome, webkit, opensourc, browser],
[refer, howto, math, latex, tutori, wiki, tabl, symbol, gwt, figur, tex],
[question, certif, java, test, scjp, mock],
[float], [db, config, configur, work],
[2011, confer, android:bookmark, hci, research, cfp, e-health],
[osx, wine, virtual, window, resourc, emul, linux], [fourthwai],
[datetim], [bookmark], [cook], [statist], [magic], [algoritmo],
[mac, perform, tcpip, congest, tech, wikipedia, tcp, network],
[taxonomi, usabl, seo, ia], [preppi], [load], [my.cnf],
[wp, wealth, wealthi, life, busi, mexico, philanthropi, person,
slim, biographi], [exam], [review],
[free, skype, voip, telephoni, phone],

.3 An example of svdCUIP, modSvdCUIP, tflidfCUIP

[bmj], [store, cheap, deal, watch, dailyd, daili],
[poll, binari, soa, backoff, algorithm, program, exponentialbackoff],
[databas, mysql, date, shell, sql, function, output, develop], [ux],
[entertain, kid, puzzl, fun, game, jigsaw], [ebm, cochran, drug],
[patient, socialnetwork, social, commun], [document], [graphic, manual],
[decis, ahp, manag, decisionmak, engin, process, econom], [write],
[powerpoint, keynot, present, design],
[informat, ehealth, internet, journal, health, healthcar, medicin, medic],
[lowcost, europ, airlin, flight, travel, vacat, hotel],
[matrix, librari, api], [ui],
[2012, lyon, public, www, www2012, via:packrati.us],
[carter], [wikibook], [interact], [new], [openoffic], [tool], [fabul],
[typographi], [mactex], [macosx], [inform_scienc], [casa], [creat],
[cs], [crian], [code], [lean], [ci], [typeset], [style], [collabor],
[whitespac], [color], [notat], [php], [tab], [tip], [spanish], [charact],
[multicriteria], [vuelo], [projectmanag], [hierarchi], [imovi], [toread],
[packag], [analysi], [command], [space], [cheatsheet], [algebra], [backup],
[train], [custom], [exampl], [lodg], [certification], [paper], [esoter],
[format], [imag], [ini], [comput], [book], [astronomi, telrad, telescop],
[osho], [certifica], [analyt],[film], [apparel], [postscript, subfigur],
[editplu, softwar, regex], [scuba, korea, dive], [firefox, extens, check],
[subscript, superscript], [fm, music, radio],
[web2.0, semant_web, elearn, forschung, educ], [iwork, appl, ilif],

.3 An example of svdCUIP, modSvdCUIP, tFidfCUIP

[mathemat, verbal, teach], [viaj, hostel, espa, hostel],
[log, overview, classif, datamin, queri], [machin-learn, ai, machinelearn],
[dna, genealog, genet, scienc, technolog, biologi],
[theater, bollywood, releas, movi, cgv],
[airport, barcelona, comida, spain, food, restaurant],
[inform, inform-scienc, li],
[restaur, vegetarian, vegan, guid],
[export, file], [video, divx, download, legenda, subtitl],
[data, excel, import, csv, financi, microsoft],
[supermercado, carrefour, onlin, compra, tienda],
[time, est, timezon, dst, convert, standard],
[financ, card, bank, creditcard, samsung]]

modSvdCUIP(k=100, d=0.63)

[[ngo], [happi, learn, epicurean, philosophi, epicuru],
[patient, socialnetwork, collabor, social, commun],
[fm, music, india, radio], [matrix, api, algebra],
[bollywood, releas, movi, hindi], [editplu, softwar, regex],
[exam, question, certif, java, test, scjp, mock, certification, certifica]
[math, mathemat, verbal, teach],
[preppi, men, wear, fashion, cloth, brand, women],
[supermercado, carrefour, casa, onlin, compra, spanish, tienda],
[financ, card, bank, creditcard, samsung],

.3 An example of svdCUIP, modSvdCUIP, tflidfCUIP

[inform_scienc, inform, ci, inform-scienc, li],
[scuba, korea, dive][time, dst], [kid, game],
[viaj, hostel, espa, hostel], [float, imag, figur],
[dna, genealog, genet, technolog, biologil],
[log, overview, classif, datamin, queri],
[video, divx, download, legenda, subtitl, film],
[wp, wealth, wealthi, life, busi, mexico, philanthropi,
person, slim, biographi],
[barcelona, spain], [openoffic], [fabul], [lodg, travel, vacat],
[data, excel, import, csv, financi, microsoft],
[free, skype, voip, telephoni, phone],
[tool, opensourc], [2011, confer, android:bookmark, hci, cfp, e-health],
[mac, wikipedia],[graphic, color, manual], [iwork, imovi, train, appl, ilif]
[perform, tcpip, congest, tech, tcp, network, linux],
[servic, search_to_rss, search, bookmark, web, rss, feed, googl],
[osx, wine, virtual, window, resourc, emul],
[librari, machin-learn, ai, machinelearn, program],
[store, cheap, deal, watch, shop, dailyd, daili],
[refer, document, howto, latex, typographi, style, typeset, whitespac,
notat, tab, tip, packag, space, command, wiki, cheatsheet, custom, symbol,
format, tex],
[algoritmo, poll, code, cs, binari, soa, backoff, algorithm,
exponentialbackoff],

.3 An example of svdCUIP, modSvdCUIP, tflDfCUIP

[firefox, extens, check],
[statist, decis, ahp, lean, manag, multicriteria, decisionmak, engin,
projectmanag, hierarchi, analysi, process, econom, analyt],
[bmj, new, informat, ehealth, journal, health, healthcar, medicin, medic],
[datetim, databas, load, mysql, date, creat, php, sql, exampl,
function, comput, develop],
[wikibook, tutori, tabl],
[web2.0, semant_web, elearn, forschung, educ],
[internet],[seo],[airport],[scienc],[research],[gwt],[paper],[food],
[hotel],[book],
[est, timezon, convert, standard],
[comida, restaurant],[2012, lyon, public, www, www2012, via:packrati.us],
[ebm, review, cochrane, drug],
[my.cnf, db, config, configur, backup, work, ini],
[powerpoint, keynot, present, design],
[astronomi, telrad, telescop],[entertain, puzzl, interact, fun, jigsaw],
[postscript, subfigur],
[cook, restaur, vegetarian, vegan, guid],
[babi, carter, crian, children, apparel],
[subscript, superscript],
[export, file],[lowcost, europ, vuelo, airlin, flight],
[theater, cgv],
[fourthwai, magic, spiritu, gurdjieff, charact, occult, esoter, osho],

.3 An example of svdCUIP, modSvdCUIP, tldfCUIP

```
[chrome, webkit, browser],[write, mactex, macosx],  
[shell, output], [taxonomi, ux, usabl, ui, toread, ia]]
```

초록

개인화 검색 및 파트너십 선정을 위한 사용자 프로파일링

변화의 비밀은 당신의 에너지를 기존 산물에 대한 비난이나 비판이 아닌 새로운 것을 구축하는데 집중하는 것이다
- 소크라테스

사용자 관심사의 자동적 식별은 도전적인 과제임과 동시에 추천 시스템에 있어 필수적이며 핵심적인 기능이라 할 수 있다. 본 학위 논문에서는, 사용자의 관심사 혹은 선호도를 식별하고 표현하는 문제를 프로파일 작성으로 치환하여 접근한다. 사용자의 관심사를 자동적으로 추론하고, 추론된 관심사 내에 잠재된 의미를 추출하는 알고리즘들을 제안하며, 제안된 알고리즘들은 개인화 검색 성능의 향상에 초점을 맞추어 고안되었다. 또한, 사용자의 프로파일을 구매자와 판매자 프로파일로 구분하여 모델링하는 방법론을 소개하며, 프로파일을 구성하는 속성들은 규정화된 용어집 (Controlled vocabulary)에 정의된 용어를 차용한다.

개인화 검색 (Personalized search) 지원을 위해 가장 먼저, Anchor text를 활용하여 사용자의 관심사를 구축하는 획기적인 방법론을 제안한다. 다음으로, 폭소노미 (Folksonomy) 시스템이 축적한 데이터에 기반하여, 행렬인수분해 (Matrix factorization) 기법을 활용, 사용자 관심사 프로파일 내의 용어 간 관계 계산을 통해 사용자 프로파일을 생성하는 두 가지 방법론이 제시된다. 제시된 두 방법론의 목적은 문맥적, 의미적 그리고 문장 구성적인 관점에서 관계를 맺고 있어 상호 그룹화될 수 있는 연관 용어들 간의 숨겨진 관계를 발견하고, 이를 기반으로 하여 용어들이 사용된 문맥을 명확히 하는데 있다 할 수 있다. 요약하자면, 사용자 관심사 모델링과 개인화를 위한 프레임워크가 제안되며, 제안된 프레임워크를 개인화된 웹 검색 관점으로 그 성능 및 유효성을 검증한다. 제안된 프레임워크를 통해 구축된 사용자 관심사 프로파일은, 프로파일의 군집화 경향 및 정확도 (Clustering tendency and accuracy) 관점에서 다시 한번 분석된다. 사용자의 질의 문맥을 정확하고 명확하게 구별할 수 있는 사용자 관심사 프로파일은, 개인화 검색 성능에 지대한 영향력을 갖는다는 것을 대규모의 실험을 통해 발견할 수 있었다.

파트너십 선정 (Partnership match)을 위해, 파트너십 온톨로지 (Partnership ontology)라 일컬어지는 온톨로지를 소개한다. 본 연구에서 소개하는 파트너십 온톨로지는, 사용자

가 자신의 요구사항들을 구매자 프로파일 혹은 판매자 프로파일로 세분화하여 지정하기 위한 초석으로 사용된다. 마지막으로, 주어진 특정 구매자 프로파일과 부합하는 판매자 프로파일들에 우선순위 할당을 위해, 의미적 유사성을 계량화 할 수 있는 지표를 정의한다.

키워드: 사용자 모델링, 사용자 관심사, 사용자 선호도, 개인화 검색, 파트너십 선정

Acknowledgements

Everyone is my teacher. Some I seek. Some I subconsciously attract. Often I learn simply by observing others. Some may be completely unaware that I'm learning from them, yet I feel deeply in gratitude. - Eric Allen

First and Foremost, I am deeply and sincerely grateful to my supervisor, Professor Hong-Gee Kim, for his continuous and instructive guidance. I thank him for his patience and encouragement that carried me on through difficult times, and also for his insights and suggestions that helped to shape my research skills and critical thinking. As an advisor, he taught me practices and skills that I will use in my future career. As a mentor, he taught me how to shape ideas into proper research and how to manage research projects from its inception to final stage. He has been a great support all along the path to PhD by supporting my teaching activities, team mentoring, proposal writing, and most important gave me a lot of freedom to develop ideas.

I am glad that I had the privilege to do my Ph.D. at Biomed-

cal Knowledge Engineering (BIKE) Lab, Seoul National University. For that also, I would like to thank Prof. Hong-Gee Kim and Prof. Myoung-Hee Kim for both supporting my Ph.D. work and establishing this unique, creative, international research environment. One of the best thing about working in this lab is that I got to work with some of the best researchers. A very conducive research environment in which collaboration with great, talented colleagues (postdocs, researchers, MS students, Ph.D. students, administrative staff) become a wonderful experience that impacted both my professional life and personal life. I enjoyed the various workshops, seminars, research meetings, project meetings, and I can proudly say that I was a part of the Biomedical Knowledge Engineering Lab (BIKE).

I would like to thank the members of my committee for their careful examination and constructive advice: Professor Hyoung-Joo Kim, Professor Sang-goo Lee, Professor Hong-Gee Kim, Professor MyoungHee Kim, Professor Im Dong-Hyuk. I sincerely thank all my fellow researchers, specially, Eung-Hee Kim, Hyun NamGoong, Seong-Jae Song, and Seong-In Lee for always being helpful over the years and making my stay in Korea a wonderful experience.

Further, I want to thank all BIKE Lab colleagues for being such a

great team and making my stay in Korea such a great experience. Last but not least, I am very grateful to my dearest son, Rik Kumar, my parents and my wife for always being there when I needed them most, and for supporting me continuously throughout these years.