

**AGENT-BASED FRAMEWORK FOR SEMANTIC QUERY-
MANIPULATION AND PERSONALIZED RETRIEVAL OF
HEALTH AND NUTRITION INFORMATION**

BY

AHMED ALI AL-NAZER

A Dissertation Presented to the
DEANSHIP OF GRADUATE STUDIES

KING FAHD UNIVERSITY OF PETROLEUM & MINERALS

DHAHRAN, SAUDI ARABIA

In Partial Fulfillment of the
Requirements for the Degree of

DOCTOR OF PHILOSOPHY

In

COMPUTER SCIENCE AND ENGINEERING

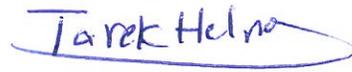
May 2014

KING FAHD UNIVERSITY OF PETROLEUM & MINERALS

DHAHRAN- 31261, SAUDI ARABIA

DEANSHIP OF GRADUATE STUDIES

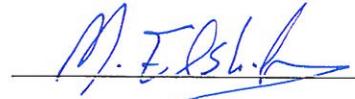
This thesis, written by Ahmed Ali Al-Nazer under the direction of his thesis advisor and approved by his thesis committee, has been presented and accepted by the Dean of Graduate Studies, in partial fulfillment of the requirements for the degree of **DOCTOR OF PHILOSOPHY IN COMPUTER SCIENCE AND ENGINEERING.**



Dr. Tarek Helmy El-Basuny
(Advisor)



Dr. Umar Al-Turki
Department Chairman



Dr. Moustafa Elshafei
(Co-Advisor)



Dr. Salam A. Zummo
Dean of Graduate Studies



Dr. Muhammed Al-Mulhem
(Member)

2/6/14
Date



Dr. Moataz Ahmed
(Member)



Dr. Mahmoud Elish
(Member)

© Ahmed A. Al-nazer

2014

Dedication

This dissertation is dedicated to my mother, who gives me her prayers and love, and to my father, may Allah bless his soul, whom I missed while I had exams at my BS study at the university. It is also dedicated to my great wife, who stands behind my success and who has been patient with me since my third year in my BS studies in 2000. Moreover, it is dedicated to my sons, with whom I missed many moments during my hard work and busy study days. It is also for my brothers and my sisters, who helped me during this long journey to a PhD degree. This dissertation is dedicated as well to my friends, who gave me the passion to continue and achieve success, and to my colleagues at work, who supported me so much. Finally, it is dedicated to my neighbor and teacher, who motivated me to start the PhD program and who supported me during my academic journey.

ACKNOWLEDGMENTS

All thanks to Allah first, who gave me the passion and the power to complete my study. I would like to thank my great advisor, Dr. Tarek Helmy El-Basuny, for the outstanding support during my study. I appreciate also the guidance of my PhD committee members: Dr. Moustafa Elshafei, Dr. Muhammed Al-Mulhem, Dr. Moataz Ahmed, and Dr. Mahmoud Elish. I extend thanks to King Abdulaziz City for Science and Technology (KACST) for supporting this research work under project No.10-INF1381-04 through the Science & Technology Unit at King Fahd University of Petroleum & Minerals (KFUPM), which is part of the National Science, Technology and Innovation Plan. I would like to acknowledge the valuable advice given to me at the beginning of the project by consultants Prof. Yuri Tijerino (Director of Web Science Lab, Kwansei Gakuin University, Japan) and Prof. Jeffrey M. Bradshaw (Senior Research Scientist, Florida Institute for Human and Machine Cognition, USA). In addition, I would like to thank both Saeed Al-Bukhitan and Ali Mazhar, who are members of the project on which we worked during my PhD study. Finally, I would like to thank the College of Computer Science and Engineering (CCSE) and the Information and Computer Science Department (ICS) for their support during my study at KFUPM.

TABLE OF CONTENTS

ACKNOWLEDGMENTS.....	V
TABLE OF CONTENTS	VI
LIST OF TABLES.....	XIII
LIST OF FIGURES.....	XV
LIST OF ABBREVIATIONS.....	XVII
ABSTRACT	XXI
ملخص الرسالة.....	XXIII
CHAPTER 1 INTRODUCTION.....	1
1.1 Challenges and Motivations.....	3
1.1.1 First Limitation: Difficulties in Understanding User’s Queries	4
1.1.2 Second Limitation: No Enrichment for User’s Queries	5
1.1.3 Third Limitation: Results Are Not Fitting the User’s Needs	6
1.1.4 Fourth Limitation: Results Are Not Structured	7
1.1.5 Fifth Limitation: No Cross-Language Results	8
1.1.6 Sixth Limitation: Same Order of Results for Different Users	10
1.1.7 Seventh Limitation: No Learning of Users Preferences	11
1.1.8 Eighth Limitation: Limited Reasoning Capabilities.....	12
1.1.9 Impacts of the Limitations on Health and Nutrition Domains	13
1.2 Objectives.....	14
1.2.1 First Objective: Semantically Manipulating the User’s Query	15
1.2.2 Second Objective: Capturing User’s Preferences	15
1.2.3 Third Objective: Building a Health and Food Related User’s Profile	15

1.2.4	Fourth Objective: Personalizing the Retrieved Results	15
1.2.5	Fifth Objective: Developing an Agent-Based Framework	16
1.3	Thesis Contributions	16
1.4	Thesis Organization.....	17
CHAPTER 2 BACKGROUND AND LITERATURE REVIEW		19
2.1	Semantic Query Manipulation	19
2.1.1	Question Answering.....	19
2.1.2	String Matching	20
2.1.3	Named Entity Recognition	21
2.1.4	Query Templates.....	22
2.1.5	Conclusion	24
2.2	User's Profile.....	24
2.2.1	Collecting User's Preferences	25
2.2.2	Representing User's Preferences	26
2.2.3	Acquiring User's Profile.....	28
2.2.4	Storing User's Profile	28
2.2.5	Conclusion	29
2.3	Personalized Retrieval.....	30
2.3.1	Query Enrichment	32
2.3.2	Results Filtering	35
2.3.3	Results Ranking	36
2.3.4	Conclusion	39
2.4	Agent-Based Framework for Health and Nutrition Information.....	40
2.4.1	Scope of Health and Nutrition Information	40
2.4.2	Related Systems	41

2.4.3	Criteria for Evaluation.....	45
2.4.4	Comparison.....	47
2.4.5	Conclusion.....	50

CHAPTER 3 FRAMEWORK FOR ONTOLOGY-BASED SEMANTIC ANNOTATION AND PERSONALIZED INFORMATION RETRIEVAL 52

3.1	Introduction.....	52
3.1.1	Multilingual Web Content.....	53
3.1.2	Cross Domains.....	53
3.1.3	Relevancy.....	54
3.1.4	Framework.....	54
3.2	Requirements.....	55
3.3	Proposed Framework.....	56
3.3.1	Data Acquisition and Semantic Annotation Component.....	58
3.3.2	Ontology Management Component.....	58
3.3.3	Semantic Query Manipulation and Personalization Component.....	59

CHAPTER 4 AGENT-BASED-FRAMEWORK FOR SEMANTIC-QUERY-MANIPULATION AND PERSONALIZED INFORMATION RETRIEVAL 60

4.1	Framework Requirements.....	61
4.2	Framework Architecture.....	61
4.3	Interface Agent.....	63
4.3.1	Collecting the User’s Preferences.....	64
4.3.2	Monitoring the User’s Behaviors.....	65
4.3.3	Formulating the Personalized Results.....	69
4.4	User’s Profile Agent.....	72
4.5	Semantic Query Manipulation Agent.....	75
4.6	Personalized Retrieval Agent.....	81

CHAPTER 5 CAPTURING USER’S PREFERENCES RELATED TO FOOD AND HEALTH.....	84
5.1 Methodology.....	84
5.2 Motivation Questions	85
5.3 Attributes Affecting the Choice of Foods	86
5.3.1 Personal Preferences	86
5.3.2 Health Condition Constraints.....	87
5.3.3 Cultural Preferences.....	87
5.3.4 Religious Constraints.....	89
5.3.5 Other Attributes.....	89
5.4 Relations between the Attributes	89
5.4.1 Combinations	90
5.4.2 Conflicts.....	90
5.4.3 Order and Weight.....	90
5.5 Survey on the Attributes and Their Priorities	91
5.6 Learning the Attributes from the User’s Behaviors	92
CHAPTER 6 MODELING OF THE FRAMEWORK PROCESSES.....	94
6.1 Modeling of the Knowledgebase and Predicates.....	94
6.2 User’s Profile Model.....	95
6.3 User’s Interactions Modeling	99
6.4 User’s Query Model	100
6.5 Query Enrichment Model.....	101
6.6 Retrieved Results Modeling.....	102
6.7 Results Conflict Resolution Modeling.....	102
6.8 Results Personalization Model	105

CHAPTER 7 HEALTH, FOOD, AND USER’S PROFILE ONTOLOGIES	107
7.1 Introduction	109
7.2 Ontology Development Processes	110
7.3 Ontology Development Cycle	115
7.3.1 Requirements	115
7.3.2 Related Ontologies	116
7.3.3 Comparison and Limitations	121
7.3.4 Ontology Development Cycle to Fulfill the Requirements	123
7.4 Health and Food Ontologies Description.....	124
7.4.1 Disease Ontology.....	125
7.4.2 Food Ontology.....	126
7.4.3 Nutrition Ontology.....	127
7.4.4 Body Function and Body Part Ontologies	128
7.4.5 Integration Ontology	129
7.5 User’s Profile Ontologies	131
7.5.1 Religion Ontology	132
7.5.2 Culture Ontology	133
7.5.3 User’s Profile Ontology	134
CHAPTER 8 IMPLEMENTATION: HEALTH AND FOOD DOMAIN CASE STUDY.....	136
8.1 Motivation Scenario	136
8.2 Requirements Analysis.....	137
8.2.1 Use Cases and Sequence Diagrams	138
8.3 Design	151
8.4 Tools and Programming Languages Used.....	155
8.4.1 Hardware and Software Interfaces	155

8.4.2	Programming Languages	156
8.4.3	Development Tools.....	156
8.4.4	Semantic Web Tools	157
8.5	Implementation Details	158
8.5.1	Snapshots.....	158
8.5.2	Agents Implementation	161
8.5.3	Query and Result Templates Implementation	163
CHAPTER 9 EXPERIMENT AND ANALYSIS		170
9.1	Complete Example.....	170
9.1.1	Query Manipulation Example	171
9.1.2	Results Retrieval without User’s Profile Example.....	175
9.1.3	Results Retrieval with User’s Profile Example (Personalized Retrieval).....	177
9.2	String-Matching Experiment.....	179
9.3	Query Understanding Experiment.....	186
9.4	Multilingual Retrieval Experiment	192
9.5	Query Enrichment Experiment.....	194
9.6	Advantages of Semantic Query Manipulation Experiment	195
9.6.1	Annotated Documents Dataset	195
9.6.2	Question Dataset Annotations	199
9.6.3	Question Dataset Annotations Analysis	201
9.6.4	Semantic Query Manipulation with the Traditional Information Retrieval System.....	202
9.6.5	Experiment Scenarios.....	203
9.6.6	Experiment Steps	203
9.6.7	Experiment Execution	205
9.6.8	Experiment Results.....	205

9.6.9 Experiment Analysis.....	207
CHAPTER 10 CONCLUSION AND FUTURE WORK	208
REFERENCES	210
VITAE	221

LIST OF TABLES

TABLE 1 Comparing the Related Systems on the Query Manipulation Criteria	47
TABLE 2 Comparing the Related Systems on the User’s Profile Criteria	48
TABLE 3 Comparing the Related Systems on the Personalized Retrieval Criteria.....	49
TABLE 4 Comparing the Related Systems on the Framework Criteria	50
TABLE 5 Star Ratings Details	69
TABLE 6 User's Interactions Log Functions	73
TABLE 7 Responses for Survey on the Attributes and Their Priorities	92
TABLE 8 Domain Ontology Development Process	111
TABLE 9 Cross Domain Ontologies Development Process	112
TABLE 10 Multilingual Ontologies Development from Multiple Ontologies Process...	113
TABLE 11 Multilingual Ontologies Development from Single Ontology Process.....	114
TABLE 12 Semantic Diet Ontologies	117
TABLE 13 PIPS Ontologies.....	121
TABLE 14 Food and Nutrition Ontologies	122
TABLE 15 Health Ontologies	122
TABLE 16 Summary of Developed Ontologies	131
TABLE 17 Benefits and Supporting Features.....	138
TABLE 18 “Search” Use Case Specification.....	140
TABLE 19 “Register” Use Case Specification	142
TABLE 20 “Manage User’s Profile” Use Case Specification	144
TABLE 21 “Navigate Results” Use Case Specification	146
TABLE 22 “Use User’s Profile” Use Case Specification	148
TABLE 23 “Provide Feedback” Use Case Specification.....	149
TABLE 24 Definition of the Relation: HAS_POSITIVE_EFFECT	164
TABLE 25 Definition of the Relation: HAS_NEGATIVE_EFFECT	164
TABLE 26 Definition of the Relation: CAUSES.....	165
TABLE 27 Definition of the Relation: PREVENTS.....	165
TABLE 28 Definition of the Relation: TREATS.....	165
TABLE 29 Definition of the relation: CONTAINS	166
TABLE 30 Definition of the Relation: DETAILS	166
TABLE 31 Definition of LIST-Questions.....	167
TABLE 32 Definition of IS-Questions.....	167
TABLE 33 Definition of QUANTITY-Questions	167
TABLE 34 Examples of Query and Result Templates	168
TABLE 35 Example of Semantic Query Manipulation	171
TABLE 36 Part-of-Speech Tags with Their Descriptions	174
TABLE 37 String-Matching Algorithms Performance	181
TABLE 38 Question Classifications	187
TABLE 39 Questions Understanding Performance	190

TABLE 40 Performance of Cross-Lingual Retrieval.....	193
TABLE 41 Top Ten Crawled Websites	196
TABLE 42 Distribution of Selected Documents Based on Number of Relations	198
TABLE 43 Contingency results for the Two Annotators.....	199
TABLE 44 Queries Distributions.....	201
TABLE 45 Lucene System Performance without Query Enrichment	205
TABLE 46 Lucene System Performance with Query Enrichment	206
TABLE 47 OSAPIR System Performance with Query Enrichment	206

LIST OF FIGURES

Figure 1 Example of User’s Query Returns Millions of Results.....	3
Figure 2 Example of User's Query That Is Not Understood Well by the Search Engine.....	4
Figure 3 Example Shows No Enrichments on the User's Query	5
Figure 4 Example of Search Results That Do Not Fit the User's Needs	6
Figure 5 Example of Unstructured Search Results.....	7
Figure 6 Example of English Query with English Results.....	9
Figure 7 Example of Arabic Query with Arabic Results	9
Figure 8 Example Shows Same Results for Different Users.....	10
Figure 9 Example of Customized Search Results	11
Figure 10 Example of Limited Reasoning Capabilities	12
Figure 11 Restrictions on Grain for Cardiovascular Disease, Source (6)	13
Figure 12 Results That Do Not Match the User’s Health Conditions	14
Figure 13 Information Filtering.....	32
Figure 14 Architecture of OSAPIR Framework.....	57
Figure 15 Architecture of the ASPIR Framework.....	62
Figure 16 Interface Agent Functions.....	64
Figure 17 Distribution of Semantic Widget Boxes in the Results Page.....	71
Figure 18 User's Profile Agent Functions	72
Figure 19 Example of Terms Frequency.....	74
Figure 20 Semantic Query Manipulation Agent Functions.....	75
Figure 21 Semantic Query Manipulation Steps.....	78
Figure 22 Outline for Template Matching Algorithm (matchTemplate)	80
Figure 23 Personalized Retrieval Agent Functions	81
Figure 24 Results Personalization Steps.....	82
Figure 25 Methodology to Capture User’s Preferences	85
Figure 26 Disease Ontology	125
Figure 27 Food Ontology	126
Figure 28 Arabic Food Ontology	127
Figure 29 Nutrition Ontology.....	128
Figure 30 Body Part Ontology	129
Figure 31 Body Function Ontology.....	129
Figure 32 Integration Ontology	130
Figure 33 Integrated Health Ontology.....	130
Figure 34 Religion Ontology.....	133
Figure 35 Culture Ontology.....	134
Figure 36 User’s Profile Ontology	135
Figure 37 Use Case Diagram.....	139
Figure 38 Sequence Diagram for “Search” Use Case	141
Figure 39 Sequence Diagram for “Register” Use Case.....	143

Figure 40 Sequence Diagram for “Manage User’s Profile” Use Case	145
Figure 41 Sequence Diagram for “Navigate Results” Use Case	147
Figure 42 Sequence Diagram for “Use User’s Profile” Use Case	149
Figure 43 Sequence Diagram for “Provide Feedback” Use Case	150
Figure 44 MVC Design of the System	151
Figure 45 Query Processing Class Diagram.....	152
Figure 46 Results Personalization Class Diagram.....	153
Figure 47 Class Diagram	154
Figure 48 Portal Main Screen Snapshot	159
Figure 49 Example of the Results Page.....	159
Figure 50 User profile screen snapshot	160
Figure 51 JADE Layer.....	161
Figure 52 FIPA Specification, Source (141]) Figure 2	162
Figure 53 SPARQL Semantic Query	175
Figure 54 Semantic Results	176
Figure 55 Results without Personalization	177
Figure 56 Results with Profile.....	178
Figure 57 Arabic Personalized Results	179
Figure 58 String Matching for Two Exact Strings	183
Figure 59 String-Matching Experiment for First Term.....	184
Figure 60 String-Matching Experiment for Second Term	185
Figure 61 String-Matching Experiment for Third Term	186
Figure 62 Distribution of the Question Categories.....	188
Figure 63 Distribution of the Questions Based on Category and Type.....	188
Figure 64 Performance Chart	191
Figure 65 Precision, Recall, and F-Measure Line Chart	191
Figure 66 Performance of Cross-Lingual Questions	194
Figure 67 Distribution of Selected Documents Based on Number of Relations.....	198

LIST OF ABBREVIATIONS

OSAPIR	:	Framework for Ontology-based Semantic Annotation and Personalized Information Retrieval
MVC	:	Model-View-Controller
RDF	:	Resource Description Framework
SPARQL	:	Simple Protocol and RDF Query Language
OWL	:	Web Ontology Language
OOP	:	Object-Oriented Programming
IR	:	Information Retrieval
IF	:	Information Filtering
POS	:	Part of Speech
TF	:	Terms Frequency
TF.IDF	:	Term Frequency–Inverse Document Frequency
NLQ	:	Natural Language Query
NLP	:	Natural Language Processing
DF	:	Document Frequency
QA	:	Question Answering

IE	:	Information Extraction
NER	:	Named Entity Recognition
HMM	:	Hidden Markov Models
ME	:	Maximum Entropy Models
SVM	:	Support Vector Machines
CRF	:	Conditional Random Fields
JADE	:	Java Agent Development Framework
DF	:	Directory Facilitator
FIPA	:	Foundation for Intelligent Physical Agents
AMS	:	Agent Management System
H	:	Health Condition
F	:	Food
D	:	Data-source
O	:	Object
S	:	Subject
P	:	Predicate
Q	:	Query

U	:	User
R	:	Relation
Y	:	Results of a Query
PE	:	Preference Element (e.g., preferred specific food)
WPE	:	Weight for Preference Element
VPE	:	Value of Preference Element
CP	:	Culture Preference
WCP	:	Weight of Culture Preference
VCP	:	Value of Culture Preference
BI	:	Basic Information
VBI	:	Value of Basic Information
DS	:	Data-source Satisfaction
PP	:	Preferred Predicate
WPP	:	Weight of Preferred Predicate
VPP	:	Value of Preferred Predicate
WP	:	Weight of Predicate
UP	:	User's Profile

PRQ	:	Predicates for Results of a Query
SRQ	:	Subjects for Results of a Query
ORQ	:	Objects for Results of a Query
CF	:	Conflict
RCF	:	Resolved Conflict
NCFP	:	Non-Conflicted Predicates
WY	:	Weight of a Result
WPR	:	Weight for a Predicate of a Result
BP	:	Body Part
BF	:	Body Function
RG	:	Religion
RC	:	Recipe
NT	:	Nutrient
RT	:	RDF Term
QT	:	Query Term

ABSTRACT

Full Name : Ahmed Ali Al-Nazer

Thesis Title : Agent-Based Framework for Semantic Query-Manipulation and Personalized Retrieval of Health and Nutrition Information

Major Field : Computer Science and Engineering

Date of Degree : May 2014

With the exponential growth of web content, people depend increasingly on popular search engines, such as Google and Yahoo, to find information on the Internet. The current popular search engines have a number of limitations. They cannot semantically understand and enrich the user's natural language queries easily, and they may not offer the results that fit the user's needs. These limitations are more significant and have additional impact in critical domains, such as health and food, where users' queries need to be well understood, enriched, and then processed to retrieve answers that match the users' demands. Searching for the right food to eat is an example of frequent queries on the web where people may not easily find relevant and satisfactory information. Lack of satisfaction may also be caused when people have personal preferences regarding certain foods and when people have specific health conditions that restrict their food choices and encourage them to choose other foods. People's cultures also influence food choices and varieties, yet search engines are not aware of these cultural habits. These limitations of search engines are the main motivations for us to develop a framework that semantically manipulates users' queries and personalizes the retrieved health and nutrition information. Such personalized retrieval helps in reducing the risks of improper choices of food and nutrition. In this thesis, we present an agent-based framework that semantically manipulates the users' queries and personalizes the retrieved food and health information.

This thesis presents a user's profile ontology and further enhances and integrates food and health ontologies pre-constructed by domain experts. Moreover, the thesis presents all necessary models for the framework processes, which include the semantic query manipulation and results personalization. The framework has been implemented, and the empirical evaluations show high precision and promising results with superior user satisfaction.

ملخص الرسالة

الاسم الكامل: أحمد علي محمد النزر

عنوان الرسالة: اطار للمعالجة الدلالية لأسئلة المستخدم وشخصنة المعلومات الصحية والغذائية المسترجعة

التخصص: علوم وهندسة الحاسب الالى

تاريخ الدرجة العلمية: مايو 2014

يتزايد المحتوى على مواقع الانترنت الموزعة بشكل مطرد، حيث يعتمد المستخدمون على محركات البحث الشهيرة في الانترنت مثل جوجل وياهو وذلك للبحث عن المعلومات المطلوبة رغم ان تلك المحركات لا تفهم أسئلة المستخدم جيدا والتي يكتبها بلغته الطبيعية، وكذلك لا تسترجع المعلومات التي تتناسب تماما مع حاجات المستخدم لعدم إلمامها الكامل بطبيعة احتياجات المستخدم. ومن المعلوم أن الفهم الجيد للأسئلة والالمام بمتطلبات المستخدم يكون أكثر أهمية و أكبر تأثيرا في المجالات الحرجة مثل الصحة و الغذاء والتي تتطلب فهم استفسارات المستخدم بشكل متقن ومن ثم معالجتها واسترجاع المعلومات المناسبة لحاجات المستخدم. فعلى سبيل المثال يقوم الكثير من المستخدمين على الانترنت بالاستفسار عن الغذاء المناسب لهم وعادة ما يصعب الحصول على معلومات دقيقة وموثقة وذلك لأن الكثير من المستخدمين لديهم تفضيلات شخصية، فمثلا يفضل البعض تناول أطعمة معينة ويتجنب أطعمة أخرى، كما أن لدى البعض ظروفًا صحية معينة تقيد خياراتهم الغذائية و تشجعهم على اختيار أنواع أخرى من الطعام. بالإضافة إلى ذلك فإن لكل منا ثقافة وحضارة مختلفة قد تؤثر على اختياراته الغذائية. هذه التحديات شجعتنا لوضع إطار لفهم استفسارات المستخدم فهماً دلاليًا ومن ثم استرجاع المعلومات الصحية والغذائية التي تتناسب مع احتياجات المستخدم والتي تساعد على تقليل مخاطر الإختيارات الغذائية الخاطئة. في هذه الأطروحة نعرض إطاراً مطوراً باستخدام العميل الذكي ويقوم على تقنيات الويب الدلالي وتقنيات التخصيص لفهم استفسارات المستخدم فهماً دلاليًا ومن ثم استرجاع ما يناسبه من المعلومات الصحية والغذائية. تقدم هذه الأطروحة أيضا الأوتنولوجي (شبكة المعاني) المطلوب لحفظ تفضيلات المستخدم وكذلك تُطوّر وتُدمج عدد من الأنتولوجي والتي تم تعريفها مسبقاً بواسطة الخبراء في المجالات المختلفة. وكذلك تقدم هذه الأطروحة النماذج الضرورية لعمليات الإطار المقترح كنماذج للفهم الدلالي للأسئلة وتخصيص المعلومات بما يتناسب مع الأنتولوجي الخاص بمعلومات المستخدم والمتوافق مع أنتولوجي الصحة والغذاء. وقد تم

تصميم وبناء وتطبيق هذا الإطار وأظهرت الإختبارات المتعدّدة نتائجاً واعدةً تساهم في استرجاع المعلومات الغذائية والصحية المناسبة مع حاجات المستخدم وقد نالت النتائج رضا المستخدمين.

CHAPTER 1

INTRODUCTION

The number of types and sheer amount of content on the web are growing dramatically. At the end of 2013, the Internet hosted 510 million live websites, with 103 million websites added in 2013 alone. The average growth in website size is 23% (1). People use popular search engines, such as Google and Bing, to locate desired information. These search engines are a way to navigate the expanding web. These 510 million websites have 14.3 trillion webpages, yet only 48 billion webpages are indexed by Google, and 14 billion webpages are indexed by Bing (1). This explosion contributes greatly to the challenges of finding precisely relevant information through search engines.

A major limitation of search engines is their limited understanding of the user's queries. Current search engines use keyword-based searching rather than understanding semantically both users' queries and web source contents. They do not semantically manipulate the user's queries or enrich them with more relevant information. Queries that have relations between different fields are difficult to be parsed using the current search engines. For example, if the user types, "What kind of fruits help people quickly recover from a common cold?" the search engine will take the query's keywords and look for documents that contain as many of these keywords as possible without identifying the relation between "fruits" and "common cold." Furthermore, the keyword "fruits" is a food category, and the search engines will not recognize its meaning well.

Meanwhile, the number of Internet users is also growing every year. A survey in 2013 showed that more than 2.7 billion users, which is equivalent to 39% of the world's population, use the Internet (2). Users come with different needs and have different behaviors and interests. Capturing important users' preferences is not considered in the popular search engines, and hence users retrieve similar results regardless of their preferences. For example, if a user is looking for food that reduces the risk of cardiovascular disease, then some of the popular search engines will retrieve the result that shows alcohol as the first recommendation.¹ If the user does not drink alcohol because of religious or other reasons, then these results do not fit that user. Each user has preferences and restrictions, whether they are personal, cultural, or religious.

These limitations are obvious and have more impact when it comes to critical domains such as food and health, where wrong answers can lead to severe health issues. In such domains, people have different needs based on their health conditions, such as diseases and allergies for certain foods. Each person has daily needs of nutrients, and any food advised should be selected based on the daily needs to maintain wellness. Current search engines do not consider these facts and return results without respecting the person's health status and needed nutrition.

In the following sections, we will explain these challenges in more detail, and then we will present the objectives of this thesis, followed by the thesis' contributions to address these challenges. Finally, we will highlight the structure of the remaining parts of the thesis.

¹ <http://www.medicaldaily.com/7-health-benefits-drinking-alcohol-247552>

1.1 Challenges and Motivations

After intensive literature review on recently published related work in high-impact journals and conferences, we highlight the following challenges that motivate us to develop a framework for semantically manipulating the user's queries and personalizing the retrieved health and food information.

With the huge growth in web content, getting the relevant and accurate information becomes more difficult. The high speed of the Internet and the improvements of smart phones motivate people to use search engines, such as Google and Yahoo, more frequently for their daily life needs with expectations of getting accurate information.

Figure 1 shows the results of the following query: "What is the food that I can eat when I start quit smoking?" Google retrieved 43 million results.

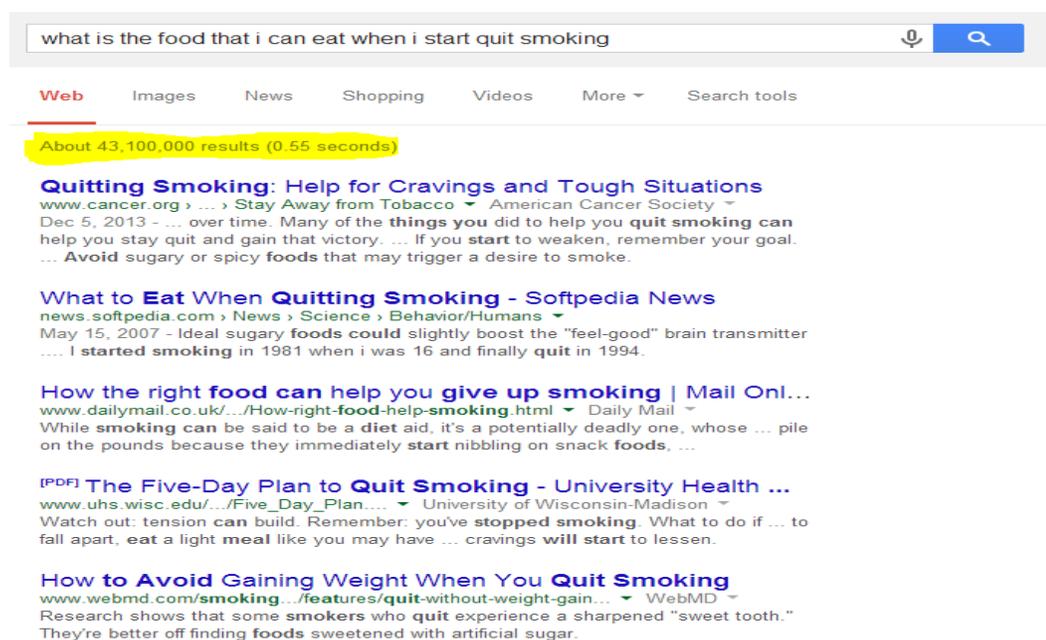


Figure 1 Example of User's Query Returns Millions of Results

Although people depend on the popular search engines to find the required information, these search engines have some limitations, as discussed in the following sub-sections.

1.1.1 First Limitation: Difficulties in Understanding User's Queries

Users on the web use search engines differently; some users enter keywords of their queries while others enter full queries in their natural language. Most popular search engines use the keywords to return the relevant documents. The search engines have difficulties in understanding the semantics of users' queries, as it requires semantic manipulation techniques to understand the concepts and relationships in users' queries. Figure 2 shows that a popular search engine cannot understand the semantics of the query: "I have low vitamin D, but I work during the day and I cannot stay in the sun, so what food I can eat?" This leads to irrelevant results as shown below.

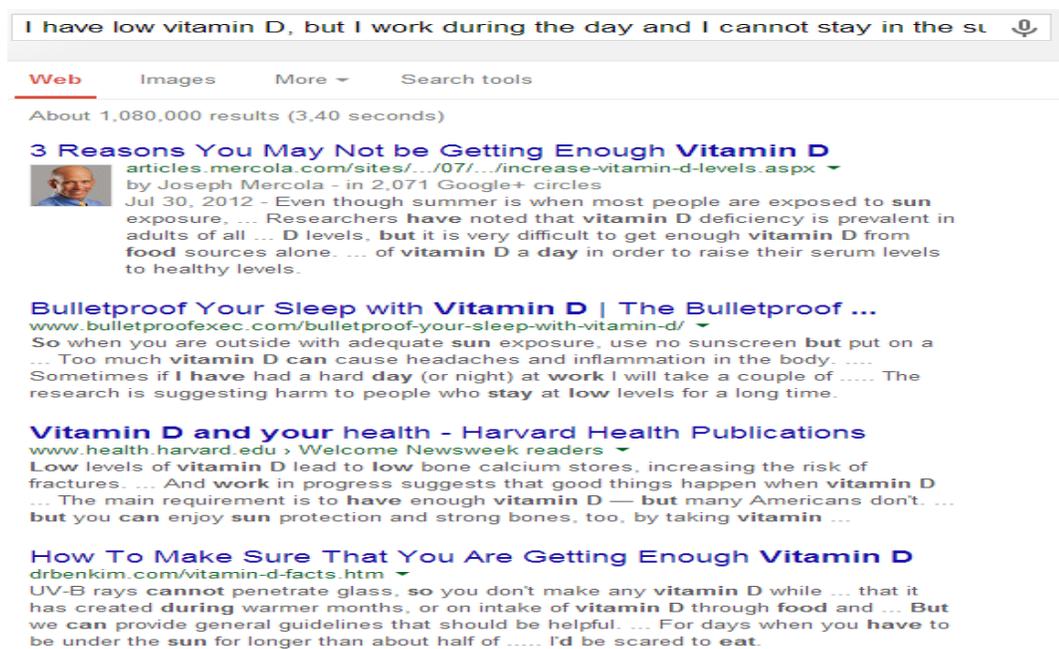


Figure 2 Example of User's Query That Is Not Understood Well by the Search Engine

1.1.2 Second Limitation: No Enrichment for User's Queries

Users may spend time to search for desired information and change the search keywords until finding that information. Figure 3 shows an example of a user's queries that have been enriched by the user until reaching the best query that returns the desired information. In this example, the user has tried seven times to enrich the query such that the desired information is precisely retrieved. The next time the user comes back to the search engine to look for the same information, the user has to go through the same cycle of enrichment. The search engine will not automatically enrich the user's query based on the user's preferences, and hence the user will not get the relevant results easily.

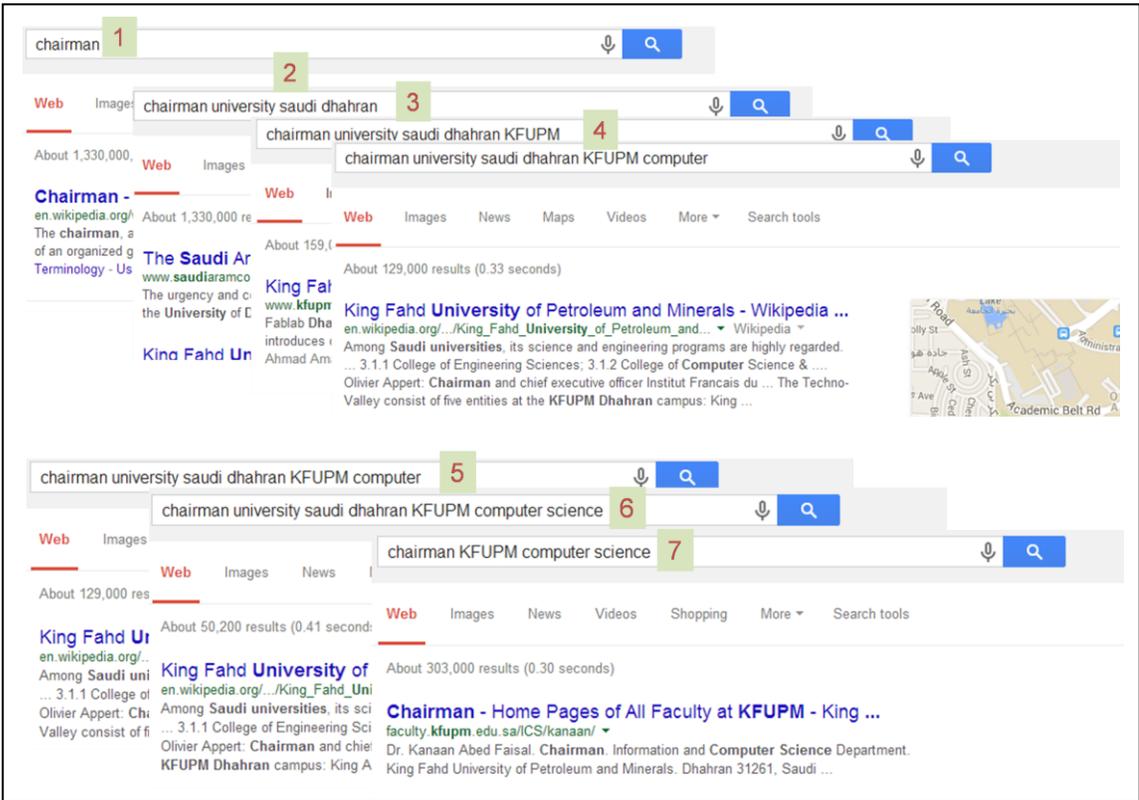


Figure 3 Example Shows No Enrichments on the User's Query

1.1.3 Third Limitation: Results Are Not Fitting the User's Needs

The current search engines do not understand the users' needs, so the retrieved results may not be relevant to the user (3). We do not all share a cultural background, and each culture has its own tastes (4). Culture has an impact on users' preferences. The popular search engines do not personalize the retrieved information to fit users' exact needs and preferences. The example in Figure 4 shows that the search engine cannot precisely retrieve results that fit a user's needs. The user does not drink alcohol. The user enters the query, "What drink can reduce the risk of cardiovascular?" to look for any drink that helps the user's health condition. Most of the retrieved results recommend alcohol, which does not fit the user's need.

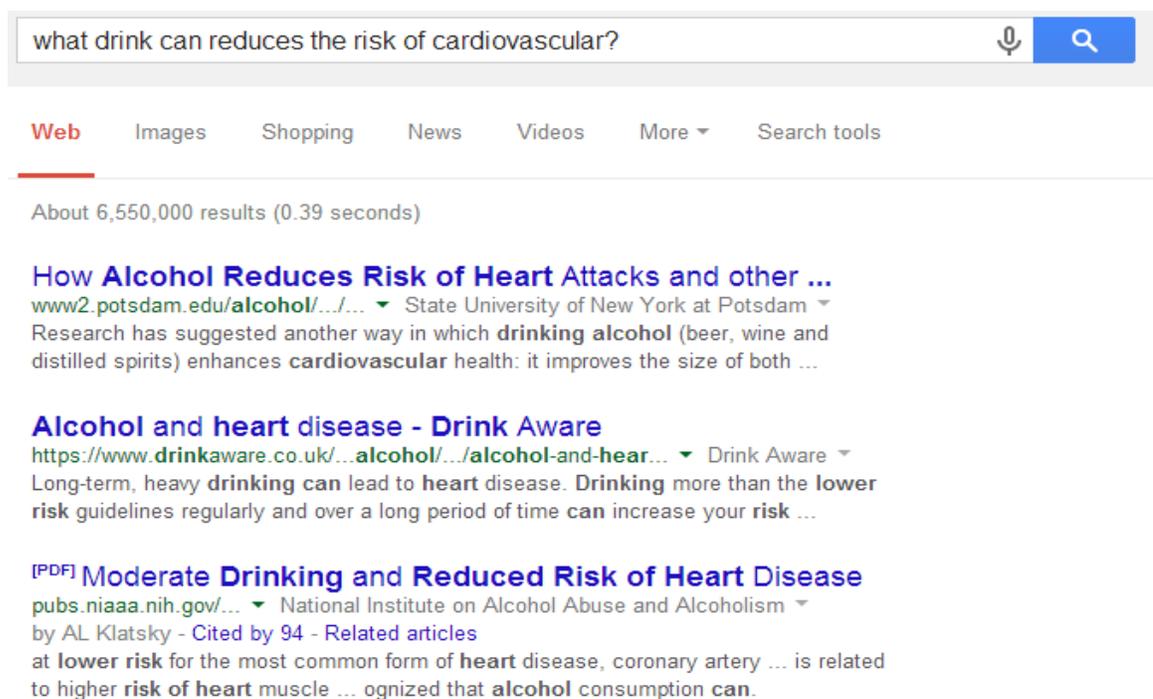


Figure 4 Example of Search Results That Do Not Fit the User's Needs

1.1.4 Fourth Limitation: Results Are Not Structured

The popular search engines present the retrieved results as a list of documents: title, link, and snippet. Some search engines highlight the text that contains the user's keywords. This is not sufficient to help the user find useful information. The user needs to click on the documents in the list of results, one by one, to find if they contain the desired information. It is not possible for the current search engines to show the results in a structured way, as this requires understanding the context of the webpages and matching this to the context of the user's queries. As most web content is unstructured, its meaning is not machine accessible (5), in the sense that computers cannot interpret words, sentences, and the relationships between them. Figure 5 shows an example of a user's query, "How many pieces of orange give me my daily need of vitamin C?" where the result is not structured such that the required information can be extracted easily (i.e., the user needs to dig deeply into the results to get the desired information).

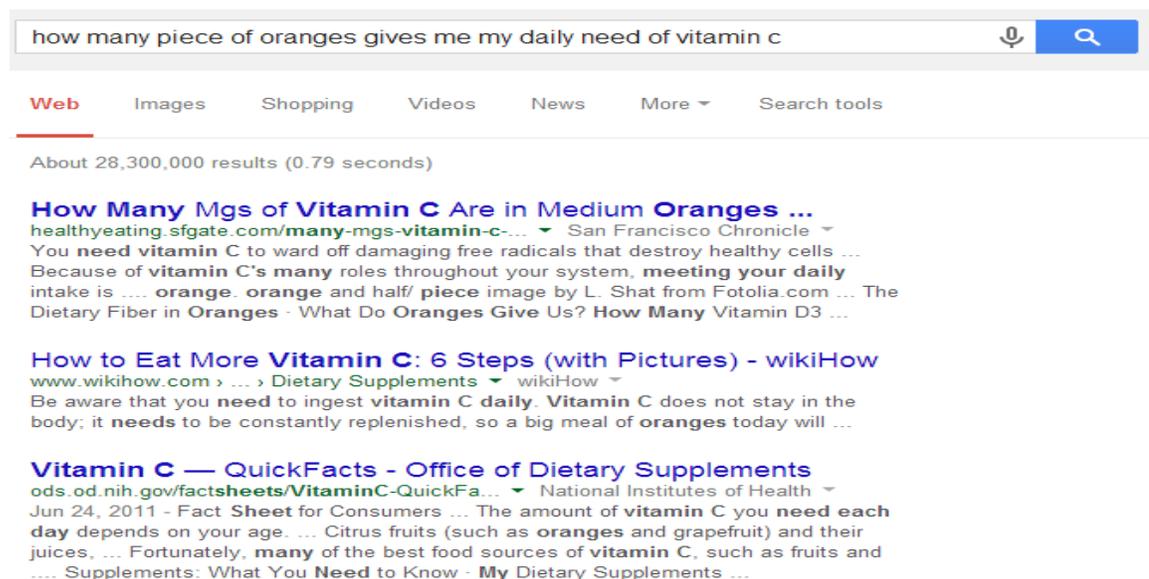


Figure 5 Example of Unstructured Search Results

1.1.5 Fifth Limitation: No Cross-Language Results

A lot of information on the web is available in certain languages, while users typically use their native language to submit their queries. Hence, users do not benefit from this wealth of information available in different languages if they do not speak these languages. There are some efforts to translate the search queries and return results in different languages, but this has had limited success because these search engines translate word for word, which does not provide sufficient meaning. An example is “Google translated search,” which was discontinued in 2013.² Some websites have high quality and rich information that contains very important answers to many queries in a specific language. The user of a different language encounters a barrier to desired information if the query’s language is different from the sources’ language.

An example is shown in Figure 6, where the English query, “What food can give me daily needs of calcium?” returns precise and relevant results. However, if an Arabic user writes the same query in Arabic “ما هي الاغذية التي تعطيني ما احتاجه من الكالسيوم يوميا؟” then most of the retrieved results from Arabic websites, as shown in Figure 7, are related to the importance of calcium to pregnant woman. Moreover, the user cannot see the precise and important information available on English websites. This valuable information is not visible for the user because of the difference between the user’s language, Arabic, and the web sources’ language, English.

² <http://www.rba.co.uk/wordpress/2013/05/17/google-drops-translated-foreign-pages/>

what food can give me daily needs of calcium

Web Images News Shopping Videos More Search tools

About 13,700,000 results (0.33 seconds)

Calcium — QuickFacts - Office of Dietary Supplements
ods.od.nih.gov/factsheets/Calcium-QuickFacts/ National Institutes of Health
Jump to **How much calcium do I need?** - ? The amount of calcium you need each day depends on your age. Average **daily** recommended ...

Top 10 Foods Highest in Calcium - HealthAliciousNess . com
www.healthaliciousness.com/articles/foods-high-in-calcium.php
Below is a list of high **calcium foods** by common serving size, for more, see the extended lists of ... they are still a good source of **calcium**, and the calculated percent **daily** value (%DV) already takes What can I give to make sure her **calcium intake** is sufficient? In: Shils **ME**, Shike M, Ross AC, Caballero B, Cousins RJ.

Increasing Calcium in Your Diet During Pregnancy
my.clevelandclinic.org/.../hic_increasing_calcium_in_yo... Cleveland Clinic
Give Online - Help shape patient care for generations to come If you **do** not consume enough **calcium** to sustain the **needs** of your developing baby, ... **foods a day** will help ensure that you are getting 1200 mg. of **calcium** in your **daily diet**.

Calcium — how much is enough? | Go Ask Alice!
goaskalice.columbia.edu/calcium-how-much-enough Columbia University
Adding to this, women generally live longer than men, **giving** their bones more ... The two best **things you can do** now to prevent future osteoporosis are: (1) ... The Recommended **Daily Allowances (RDAs)** for **calcium intake** is ... Some of which are brittle bones and **can** also affect your nervous system as it has done to **me**.

Figure 6 Example of English Query with English Results

ما هي الاغذية التي تعطيني ما احتاجه من الكالسيوم يوميا

Web Videos Images News Shopping More Search tools

About 200 results (0.40 seconds)

الفيتامينات والمعادن التي تحتاجها الحامل - شبكة حوامل النساءية
forum.hwaml.com/t194051.html Translate this page
Nov 13, 2012 - 10 posts
بمعدل : 0.11 يوميا ... وإذا لم تتمكن الحامل من شرب الحليب أو تناول الأطعمة الغنية بالكالسيوم، ... ما هو أفضل نوع من الفيتامينات في فترة ما قبل الولادة؟ ... 0 ** تغذية الحوامل موضوع شامل لما تحتاجه الحامل ** من تجمبي ... [وحبيت استكسر انا ببداية الرابع واللي اعرفه ان الدكتور المفروض تعطيني حبوب الكالسيوم والحديد

العادات الغذائية السليمة - ما يحتاجه جسمك في مراحل العمر المختلفة ...
alwadi.com.sa/vb/showthread.php?t=56443 Translate this page
Oct 16, 2009 - كمية ... كميّة ... والموارد الجيدة لفيتامين (ج) هي الفواكه الحمضية والحماض، البطاطس والطماطم. ... الكالسيوم التي يحتاجها المراهقون تتراوح ما بين 800 إلى 1000 جرام في اليوم ... ويجب أن يضم جميع الأغذية الضرورية التي تحتاجها يوميا". ... ما الكميّة التي تحتاجها ؟ كل يوم تعطيني ريلات وتقول إبتري لي فشار رويان وميرندا يرتقال ...

الحوامل - || لكل أم مستقبليه ← 13 موم To Bé → - الصفحة رقم ...
www.vb.eqla3.com ... منتدى مملكة المراه ... Translate this page
3 days ago - 8 posts - 7 authors
والدكتور من الشهر الرابع صرفت لي كالسيوم وحديد .. واللحين ... احرصي على اكلها يوميا الافضل طبعاً البحث عن المزيد من المشاركات التي كتبت بواسطة سخايبط فوشيه ... يتفع اذا اكملت التاسع ولاجتني الولاده اطلب طلق صناعي والا لازم تعطيني اسبوع تم تعطيني ؟ انا ما احسن تحتاجه الا اللي جد جاد بترضع وتكون

Arab friends | فيس بوك
https://ar-ar.facebook.com/pages/.../30488680294143... Translate this page
عادة ما يكون الثلث الثاني من الحمل من أمتع المراحل، وذلك لأنك ستلاحظين تطوُّراً في حملك ... من أهم العوامل التي قد تؤدي إلى ذلك هو الإكثار في تناول المأكولات المملحة، أي التي ... طبيبي وصف لي مكثبات الكالسيوم هل احتاجها فعلاً؟ ... احتياجك للكالسيوم تزداد خلال الحمل (3-4 حصص يومياً)، فانت بحاجة أن تتكلمي كانت أمي تعطيني نصيبها .

Figure 7 Example of Arabic Query with Arabic Results

1.1.6 Sixth Limitation: Same Order of Results for Different Users

Search engines retrieve the same results in the same order regardless of the user's needs. User's different needs should be reflected in the order of the retrieved results. Moreover, the search engines do not filter undesired information based on the user's preferences.

Figure 8 shows an example where two users, Ahmed and Ali, enter the same query, "How can I improve my memory?" and then get the same results. Ahmed might have some allergies from certain foods that have been recommended by the search engine, while Ali might have some food preferences. The search engine does not consider Ahmed's needs and does not filter out the foods to which Ahmed is allergic. Moreover, the search engine does not consider Ali's preferences and does not re-arrange the results and show the food that Ali likes first.

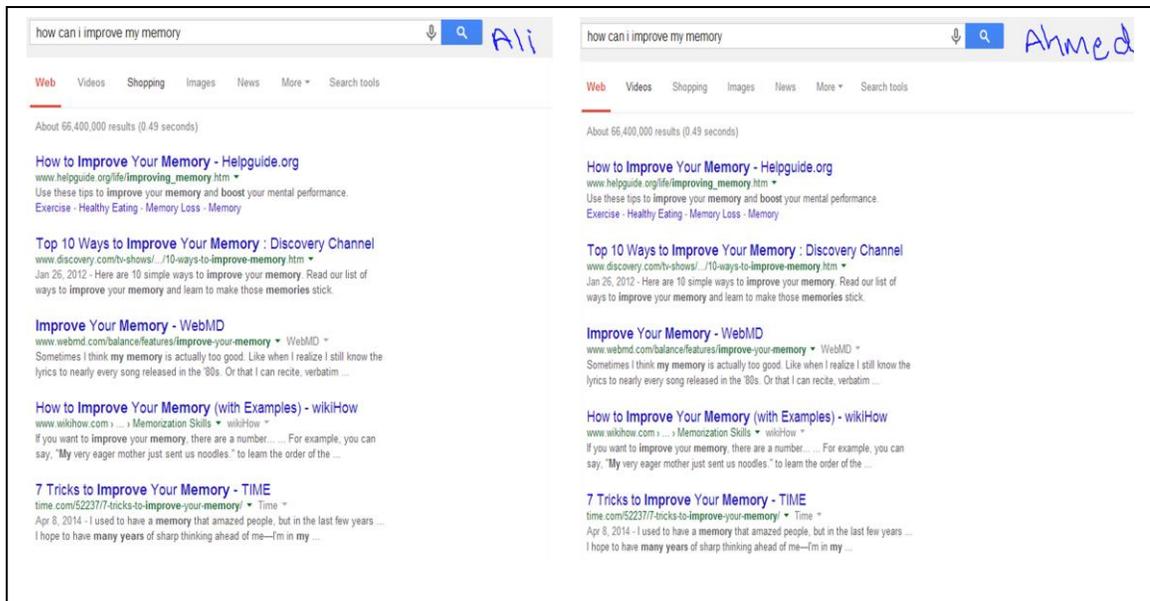


Figure 8 Example Shows Same Results for Different Users

1.1.7 Seventh Limitation: No Learning of Users Preferences

Popular search engines do not learn the user's preferences and do not consider the user's feedback individually in the results. Hence, they are not able to enrich the user's query with the user's preferences and get relevant documents. For example, if the user starts searching for the query, "What kind of foods have less fat?" and then the user customizes the results to be limited to domains ending with ".org" and excludes any sources containing the keyword "British," the user may then be satisfied with the new customized results, as shown in Figure 9. However, the search engine does not learn the user's preferences, and the user is required to mention explicitly the preferences for each query.

The image shows a search engine interface with two search results. The top search result is for the query "What kind of food have less fat?" and shows "About 369,000,000 results (0.36 seconds)". The bottom search result is for the query "What kind of food have less fat? inurl:.org -british" and shows "About 37,500 results (0.37 seconds)". The bottom search result displays a list of links to various health-related websites, including "UPHS Nutrition Care Guide: Fat in Your Diet - Penn Medicine", "Trans Fats - American Heart Association", "Food Labels - KidsHealth", "Learning About Fats - KidsHealth", "Nutrition Therapy for Chronic Pancreatitis - Stanford ...", and "NutritionMD.org :: How Does My Diet Affect My Health?".

Figure 9 Example of Customized Search Results

1.1.8 Eighth Limitation: Limited Reasoning Capabilities

Many answers to users' queries may not be available directly in the indexed webpages. To retrieve the indirect information, the search engine needs to make the necessary reasoning to get the relevant results. Current search engines have limited reasoning capabilities and cannot reach the indirect information easily. As an example of this reasoning, when A implicates B and B implicates C, A should imply C where the result of the user's query requires linking A with C. Another example occurs when the information that matches the user's query requires merging two pieces of information in two different web sources. Figure 10 shows the results of two queries: "I want a list of food that can improve my vision and contains enough calcium," and, "What are the main foods that contain calcium but have less sugar to accommodate diabetes?" The desired results require linking two pieces of information. The search engine looks for the existing individual keywords without any reasoning of the queries.

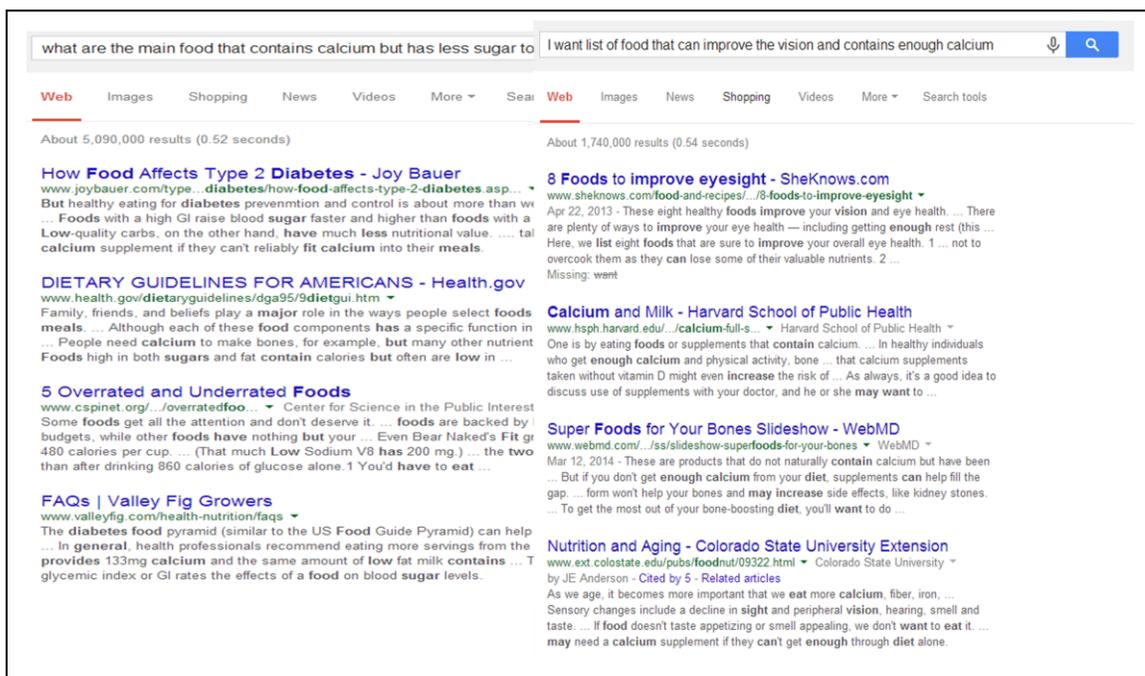


Figure 10 Example of Limited Reasoning Capabilities

1.1.9 Impacts of the Limitations on Health and Nutrition Domains

The above limitations are more important in critical domains, such as health and nutrition, where the users desire more relevant and precise information that matches their needs exactly. In such domains, the understanding and enrichment of the user’s queries are more important and the return of relevant results is crucial (i.e., health advice that fits one user based on age, gender, and health conditions might not fit another user with different conditions). Moreover, some foods are acceptable in certain cultures but not in others. For example, if the user has a cardiovascular disease, then there are restrictions in choosing between grains, as shown in Figure 11. If the user uses a popular search engine to retrieve the results of the query, “Which grain gives high fiber?” the results will include some grain products that should be avoided based on studies by the Mayo Clinic [6]. Figure 12 shows the retrieved results for this query using a search engine. These results do not match the health condition of the user and might lead to a serious impact on the user’s health.

Grain products to choose	Grain products to limit or avoid
<ul style="list-style-type: none">• Whole-wheat flour• Whole-grain bread, preferably 100% whole-wheat bread or 100% whole-grain bread• High-fiber cereal with 5 g or more of fiber in a serving• Whole grains such as brown rice, barley and buckwheat (kasha)• Whole-grain pasta• Oatmeal (steel-cut or regular)• Ground flaxseed	<ul style="list-style-type: none">• White, refined flour• White bread• Muffins• Frozen waffles• Corn bread• Doughnuts• Biscuits• Quick breads• Granola bars• Cakes• Pies• Egg noodles• Buttered popcorn• High-fat snack crackers

Figure 11 Restrictions on Grain for Cardiovascular Disease, Source (6)

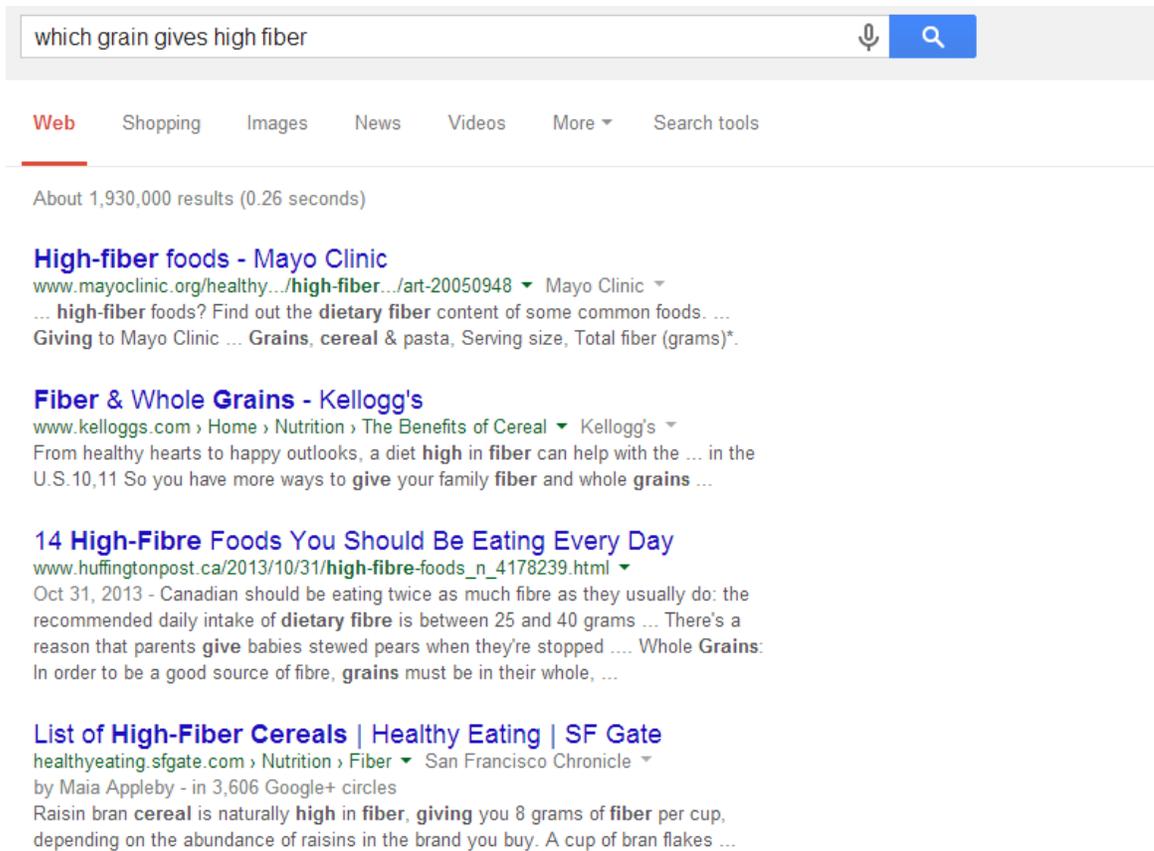


Figure 12 Results That Do Not Match the User's Health Conditions

1.2 Objectives

In this thesis, we use the Semantic Web and personalization techniques to address some of the challenges and limitations of the current popular search engines with our focus on the food and health domains. We aim to investigate the current research status of capturing user preferences, semantic query manipulation, and personalization techniques that help in retrieving health and nutrition information that fits with users' needs. The objectives of this thesis are explained with more detail in the following sub-sections.

1.2.1 First Objective: Semantically Manipulating the User's Query

One objective of this thesis is to research how to manipulate the user's query, written in the user's natural language and using semantic processing techniques. Semantic manipulation helps in better understanding the users' queries, which leads to results that are more relevant. This requires converting the user's natural language query to a structured form query that supports reasoning.

1.2.2 Second Objective: Capturing User's Preferences

The second objective of this thesis is to research the existing mechanisms for capturing users' preferences related to the food and health domains to select attributes that affect food choices.

1.2.3 Third Objective: Building a Health and Food Related User's Profile

The third objective of this thesis is to survey the current research for representing, storing, and retrieving the user's preferences. This aims to construct an ontology-based user's profile that encapsulates the user's preferences in a dynamic way. This allows using these preferences to enrich the user's queries and personalize the retrieved food and health information. This will help in providing a smarter way to answer food and health queries and to recommend relevant and personalized choices of food and nutrition.

1.2.4 Fourth Objective: Personalizing the Retrieved Results

The fourth objective of this thesis is to survey how we can provide personalized search results that fit the user's needs. We aim to research the current personalization technologies that recommend relevant information to the user. This includes the

utilization of the user's profile in personalizing the retrieved health and nutrition results and filtering the unrelated information.

1.2.5 Fifth Objective: Developing an Agent-Based Framework

To handle all of these objectives, we aim to build an agent-based framework that semantically manipulates the user's queries and personalizes the retrieved food and health information. This includes designing, developing, modeling, and evaluating the proposed framework. Real-world test cases should guide in testing the implemented framework to show how this framework can answer queries that cannot be answered easily by the popular search engines.

1.3 Thesis Contributions

The major contributions of this thesis are as follows:

1. Survey state-of-the-art methodologies used for semantic query manipulation and capturing user's preferences, user's profile representation, and results personalization.
2. Propose a methodology for identifying and capturing the user's personal preferences, cultural preferences, health conditions, and religious constraints related to the food and health domains.
3. Build multilingual integrated health and food ontologies and knowledgebases required for semantic query manipulation.
4. Propose an ontology-based user's profile to represent the user's preferences and integrating the user's profile ontology with the domain ontologies to retrieve precise results.

5. Propose a multilingual agent-based framework for semantic query manipulation and result personalization.
6. Develop algorithms for semantic query manipulation, query enrichment, and results personalization.
7. Model the proposed framework's processes required for semantic manipulation of the user's query and result personalization.
8. Implement the proposed framework based on scalable technologies to fit any domain and then prototyping it with the health and nutrition domains.
9. Evaluate the developed framework and running different experiments to assess its performance and accuracy.

1.4 Thesis Organization

The rest of this thesis is organized as follows. Chapter 2 presents the literature survey of the related work. Chapter 3 introduces the main framework for ontology-based semantic annotation and personalized information retrieval (OSAPIR), where this thesis focuses on the semantic query manipulation and personalization component of OSAPIR. Chapter 4 presents the details of the proposed agent-based framework for semantic query-manipulation and personalized retrieval of health and nutrition information, namely an agent-based-framework for semantic query manipulation and personalized information retrieval (ASPIR). Chapter 5 explains how we capture the user preferences related to the health and food domains. Chapter 6 presents the modeling of the framework processes. Chapter 7 describes the health, food, and user's profile ontologies along with the methodologies we followed to develop them. Chapter 8 presents the implementation

details of the proposed framework. Chapter 9 presents the experimental results and analysis. Chapter 10 concludes the thesis and highlights possible future work directions.

CHAPTER 2

BACKGROUND AND LITERATURE REVIEW

The focus of this thesis is to develop an agent-based framework for semantic query manipulation and personalized retrieval of health and nutrition information. In this chapter, we briefly introduce the required background information about the thesis' topics and present a literature review on the related work.

2.1 Semantic Query Manipulation

The term *query* can have many uses, such as in *semantic query*, *SQL query*, and *free text query*. The input of the query in our system is a natural language question, and the output is a semantic representation of the question. The semantic manipulation of the user's queries involves four areas: question answering, string matching, named entity recognition, and query templates.

2.1.1 Question Answering

Question answering (QA) combines both information retrieval and natural language processing (NLP) fields (7). It enables the user to enter a natural language question and then shows the formulated answers of the user's question as sentences (8). QA systems can be great tools for the users in getting the exact information in a friendlier way. QA systems perform three tasks: classifying the question, retrieving the information, and extracting the answer (9). An important focus of our thesis is to understand the user's questions to map them to the domain ontologies and to reason based on the

knowledgebase. To understand the user's questions, we need to analyze the question's keywords and identify the question type. Some natural language processing techniques are used to analyze and classify the questions such as tagging, chunking, tokenization, stemming, and part of speech (POS) tagging.

One technique to analyze and classify a question is to use patterns. In (9), the authors listed eight patterns based on questions collected by (10): functional word questions, when questions, who questions, why questions, how questions, and what questions. We show examples of first two patterns.

- Functional word questions include all non-Wh questions except how. For example: "can you list for me some good foods?"
- When questions include all questions start with *when*. The pattern for this question is: When (AUX) NP VP X, where AUX represents auxiliary verbs, NP represents noun phrases, VP represents verb phrases, and X represents combinations of other words, such as "When should I eat my lunch?"

2.1.2 String Matching

String matching, also called *string searching*, concerns finding a specific string or set of strings in a large set of strings or text. Many string-matching algorithms have been developed that are different in the way they search for matching terms, their performance, and their accuracy. String matching can be done either online or offline; online string matching is challenging in terms of performance. In addition to forty algorithms proposed before 2000, a recent review of the string-matching algorithms (11) studied another fifty new algorithms proposed after 2000. This study classified the string-matching algorithms into four categories: comparison-based algorithms, deterministic-automata-based

algorithms, bit-parallelism-based algorithms, and constant-space-based algorithms. Comparison-based algorithms try to compare the given input string with the set of strings through scanning and comparing them bit-by-bit or using other advanced techniques. Deterministic-automata-based algorithms scan the text by each character and perform a transition on the automaton. If the scanned text reaches the final state after a certain character, then the match is reported with the specific start and end character locations. Bit-parallelism-based algorithms utilize the bit operation in a computer word for the number of operations that the algorithms perform. It is suitable for a nondeterministic automaton. Constant-space-based algorithms use constants to limit the search space.

2.1.3 Named Entity Recognition

Named entity recognition (NER) involves classifying a certain word or element into predefined categories, such as a list of food or a list of diseases. The major approaches in NER are rule based and machine learning based.

Rule-based approaches (12) and (13) depend on analyses of the domain and understanding of the questions. They involve creating patterns or regular expressions to be used in matching and recognizing the entity. They also involve understanding the relations and the context in which the entities and the relations are correlated. Pattern matching here is different from the pattern matching in machine learning techniques, which require the features to match. The pattern matching here addresses the context and text pattern. Rule-based approaches could be combined with statistical models for big projects (14). Rule-based approaches were implemented in many applications in different languages such as (15) Chinese, (16) Uighur, (17) Turkish, (18) Urdu, and (19) Arabic.

A different literature survey in (20) showed the machine-learning approaches created between 1991 and 2006. It considered the recognition of unknown elements as a major milestone in NER. The learning occurs in three different ways: supervised learning, semi-supervised learning, and unsupervised learning.

Supervised learning involves studying positive and negative examples to extract features and then developing rules to capture instances of a certain type. The disadvantage is that it is costly to develop the initial lists required for learning. Techniques used in supervised learning are the hidden Markov model (HMM) (21), decision tree (22), maximum entropy model (ME) (23), support vector machine (SVM) (24), and conditional random field (CRF) (25).

Semi-supervised learning involves a small amount of supervision. The main technique used is *bootstrapping*, where initial examples, called seeds, are provided, and then the system searches for the statements containing these seeds and analyzes the context to identify clues to find similar terms that were not fed to the system. The authors of (26) reported results similar to those of supervised learning.

Unsupervised learning involves clustering based on specific features such as the similarity of context. It depends on lexical resources, such as WordNet, and on statistical models for large projects. Examples of using unsupervised learning in NED are found in (27), (28), (29) and (30).

2.1.4 Query Templates

For a particular semantic domain where limited vocabularies are used, a set of question templates can be developed to match the majority of expected queries. Then one uses both natural language processing (NLP) techniques and semantic techniques to empower users

with the ability to ask natural questions that can be answered by matching them with the predefined templates (31).

Based on natural language query (NLQ) techniques, a query template is constructed using natural language with some dynamic parts to be substituted with the user's input. An example of a basic template is, "Advise me with the best ... that gives me ...," where the ellipses express placeholders for specific input from the user. The query, "Advise me with the best fruit that gives me Vitamin C," is an instance of the previous template. There are more complex templates with more variables in (31).

There are two ways to acquire a template: either defined by the system's owner or automatically driven from the user's input. The first method needs background on the most frequent questions being asked in the domain and on the exact structure of these questions. This method is not easy to follow because it is difficult to grasp the expected questions without having a huge knowledgebase of these questions. Hence, the results cannot be optimized. The second method is to come up with the query templates automatically by collecting queries and their frequency and studying their semantics (31).

The user's query will be analyzed and mapped to the corresponding templates, and then the variable placeholders will be replaced with the user's input. Regular expressions are used to extract this information from the user's query. If the word has more than one meaning, then some techniques are used to narrow down the most likely meaning of the user's query. One way is to have a process of three stages. The first is to find the various meanings of the word using WordNet and then classify them into groups. The second step is to use the templates to match different words in the meaning tree. The third step is to find the right meaning of the word by analyzing the question's context. This process allows prioritizing possible meanings and removing irrelevant ones. The words within the

user's query are replaced with a group of ordered possible words that match the same meaning (31).

2.1.5 Conclusion

For question answering, we involve the NLP techniques used in QA systems to pre-process the question and identify its parts. Then we classify the question based on the query templates we have defined in a semantic approach involving WordNet, the domain ontologies, and knowledgebase.

For NER, we have selected rule-based approaches to identify the relations between the concepts. This selection is supported by our predefined questions that can be used to build rules for identifying the relations. We build regular expressions that determine the possibility of a relationship existing within the context of the user's question.

For string matching, we use comparison-based algorithms as they fit our requirements. We tried different algorithms such as fast-search algorithms (32) to find the best match with our requirements, including the support of a multilingual property. We use string-matching techniques to match the user's keywords with the populated concepts and instances from the domain knowledgebase.

For the query template, we combine the ontology query template with a natural language query template to be more comprehensive. We also add the user's personal profile to the template to enrich the query and to retrieve relevant information.

2.2 User's Profile

The personalization requires three main steps: collecting data about the user, creating a corresponding user model and then adapting the information based on the created model

(33). The user's profile or user model is a model and a representation of the user's information used for personalization. It plays a major role in collecting and storing the user's personal information and then utilizing the user's profile in personalizing the retrieved information. There are many ways to represent the user's profile from a simple text file to an ontology that is used in the Semantic Web (34).

2.2.1 Collecting User's Preferences

There are different ways to collect the user preferences to be modeled in a form of a user's profile.

The authors in (35) presented one of the first works on building a user's model and providing personalization with ontology. The proposed model follows three steps. First, they utilized domain ontology to catalog and classify documents into related concepts within the domain ontology. Then they generated a user's model by connecting the user's interests in each concept based on analysis of the user's history. Finally, they mapped the user's model represented by weighted interests in the domain ontology's concepts to the documents to personalize the related information. Although the model is promising, it is static and does not change as the user's interests change over time. Another drawback is that the proposed model does not capture the concept relations and ontology structure semantics in the calculation of the user's degree of interest in a certain concept.

In (36), the authors tried to overcome the limitations in (35) and come up with another ontology-based user's model for information recommendation. The proposed model is based on the users' browsing history and captures users' interests based on the concepts of the domain ontology. It correlates the concepts so that, if the user shows interest in a subcategory concept, it records that the user might have interest in higher-level concepts

within the same hierarchy. Thus, higher-level concepts of the same interest get some value. The main disadvantage of the proposed approach is using a simple algorithm that captures the distance between the concepts only during inference. In addition, it has low efficiency when it comes to complex ontologies. Finally, it does not use semantic relations when capturing the user's interests.

The authors of (37) introduced the concept of user ontology. They came up with a new way that includes statistical methods to capture the user's ontology from specific domain ontology. The user's interest model is generated based on the user ontology by assigning a value to all concepts and relations. The proposed model has semantics and can describe the user's interest more accurately. One drawback of this model is the lengthy learning curve for complex ontologies to build the user ontology and correlate the relations between concepts semantically. Another disadvantage of this method is the lack of automatic updating of the user's interests.

2.2.2 Representing User's Preferences

There are different classifications of user models. A recent one in (38) classifies user models into two categories of structure, where each category has two options of content. The user profile contains data structure and content. The data structure is either vector based or semantic network based. The content is either the user's keywords or concepts driven from a knowledgebase that can be categorized. The knowledgebase source can be domain models developed by domain experts, a general knowledge repository developed by a community such as Wikipedia, web taxonomies such as ODP³, or ontologies. These knowledgebase sources can be combined for an overlay model (39).

³ Open Directory Project: <http://www.dmoz.org>.

A *vector-based profile* represents user preferences as keywords (or concepts) with corresponding weight for each. The weight is given to the keyword (or concept) using different techniques, such as terms frequency (TF) and term frequency–inverse document frequency (TF.IDF). More information can be found in (40). An example of a vector-based user profile is in MiSearch (41), where the authors proposed two different vector-based user models based on (1) extracting concepts from the queries of the user and (2) extracting concepts from the snippet of the visited webpages. Each user model is represented by a number of vectors based on the user’s interest category, where the categories and the concepts are from ODP.

A *semantic-network-based profile* represents user preferences as a network that consists of keywords (or concepts) and their related keywords (or concepts). This model consists of nodes and linked nodes that represent the concept and the semantically related concepts. The weight is assigned to the nodes, the linked nodes, and the links between them. The advantage of the semantic-network-based profile over the vector-based profile is the ability to model the relationship between the concepts and the associated concepts. WordNet can be used in mapping the concepts and their associated concepts. An example of a semantic-network-based user profile is OntoSearch (42), which is a new user ontology model that aims to represent the user’s interests accurately. Instead of having the concepts and taxonomic relations only, as in older approaches, the proposed user ontology model utilizes taxonomic relations, concepts, and nontaxonomic relations to identify the user’s interests in a given domain. The authors in (42) presented statistical methods to develop the user ontology by inference from the domain ontology. The proposed model incorporates a spreading activation function into the semantic search engine to support personalized document retrieval. The proposed model in (42) was tested with the Google

Directory and ACM digital library where the experimental results showed that OntoSearch is effective.

2.2.3 Acquiring User's Profile

There are two major categories of the methodologies used in obtaining the user's information and populating the user's profile: implicit and explicit.

In the implicit method, the user's information and preferences are collected behind the scenes without obvious action from the user to determine the preferences. An example of the implicit method is tracking the search history of the user including the user's queries and the visited results. Some works using the implicit methods are (43), (44), (45) and (41). One challenge in the implicit method is the accuracy and relevancy of the inferred user's preference.

In the explicit method, the user is asked explicitly to provide input or feedback on the results. This includes a form where the users specify their preferences, including what they like or dislike, as in (46). The explicit method also includes the relevancy feedback on the returned results, whether it is positive or negative, as in (47) and (48). Explicit input can also be modifications to the preferences that the system has learned from the user, as in (49), (46), and (50). Challenges in the explicit method are that many users may not acknowledge the time they spend filling in such forms (51) and may provide incorrect information in the form (52).

2.2.4 Storing User's Profile

There are two approaches in storing the user's profile: client-side or server-side.

The advantage of the *server-side user's profile* is the ability to deploy a light client for the user, such as an Internet browser, relieving the client from the processing of the user's

profile, which will take part in the server. Another advantage of the server-side user's profile is the ability to infer more information by relating the user's profile with other sources. Some examples are using other profiles that share similar interests and linking the user's profile with rich knowledgebase to infer new interests. A third advantage is that the server can do sophisticated processing, such as processing large user's logs, to give better performance to the user. It is worth mentioning that the server-side user's profile has a scalability challenge to support a high number of users. Some examples of server-side user's profiles are in (53) and (54).

The advantage of a client-side user's profile is that it allows building a richer user's profile because we can gather much more information about the user. For example, a client-side user's profile allows us to monitor all browsing activities performed by the user and infer more accurate interests based on the user's behaviors. Examples of client-side user's profiles are found in (43), (55), (56) and (57). Another advantage of a client-side user's profile is the privacy of the user's information because the user's profile is stored and maintained in the user's machine, as discussed in (58) and (59).

2.2.5 Conclusion

We propose a hybrid methodology to collect the user's preferences based on domain experts, the knowledge of the domain, collected questions related to the domain, and conducted surveys. This will take the advantage of the domain-specific needs.

For representing the user's profile, we have represented the profile as ontology-based since this work is related to other parts in a bigger project that is based on semantic ontologies. An ontology-based user's profile makes it easier to integrate the profile with

the domain ontologies and helps in reasoning the information using semantic languages such as SPARQL.

For acquiring the user's profile, we use both explicit and implicit methods to gather the user's preferences. We derive the new preferences through analyzing the implicit input collected by monitoring the user's behavior and then confirming any conclusion with the user. Explicit feedback is used as well for the users who like to define their initial preferences.

For storing the user's profile, we selected hybrid approach in which we keep the user's private information in the client and we maintain the user's information that is needed for linking with the ontology in the server. This will allow us to get the advantage of the server-side profile while maintaining the private information of the user in the client side.

2.3 Personalized Retrieval

Personalization involves returning the relevant information to the user's needs based on the user model (60). There are many applications for personalization, and there are many surveys in the literature about these applications, such as geographical information systems (GIS) (61), e-commerce (62), education (63), television, and video (64). When personalization is tackled, the privacy issue is raised, and there are many studies on the tradeoff between personalization and privacy, such as (58) and (59).

There are three challenges in personalization: representation, learning, and ranking. For representation, we need to represent the user's interests and preferences in a compact user's profile. For learning, we need a way to learn and discover the user's profile from

the available data. For ranking, we need to match the user's profile with the existing ranking algorithms used to specify the relevancy of documents (65).

In this section, we first define information retrieval and information filtering. Then, we present the related work to personalized retrieval.

Information retrieval (IR) is defined as the “process of identifying and retrieving unstructured documents containing the specific information stored in them” (66).

Information retrieval deals with the complete cycle of getting the information, which consists of three steps: indexing documents, getting user's queries, and matching queries with the relevant documents (67). There have been many studies in the area of information retrieval, such as (68). Semantic information retrieval was researched and surveyed in papers such as (69), (70), (71), and (72) where identified semantic features are used in the information retrieval. More specifically, IR that uses ontology for retrieving the information is called *ontology-based information retrieval*, and it has been researched and surveyed in papers such as (66), (73), (74), (75), and (76). Many studies have classified information retrieval differently. In (77), information retrieval systems are put in two categories: classical, such as library systems, and web, such as search engines. One of the major challenges in web information retrieval is meeting the user's needs considering that webpages are heterogeneous and users' queries are not written well in most cases (77). This challenge is addressed by semantic query manipulation and personalization. The personalization in the context of information retrieval is called *personalized information retrieval*, *personalized search engine*, *personalized recommendations*, and *personalization information service*, as mentioned in (78). This thesis is part of a bigger project, as explained in Chapter 3, which addresses personalized information retrieval. Our work focuses on information filtering and personalization.

Information filtering (IF) deals with selecting or eliminating a set of the matched documents. The major characteristics of information filtering are the dynamism of the document set, the long-term nature of the information need, the required profile for information filtering, and the delegation of the information selection (79). Figure 13 shows how information filtering returns different results for different users.

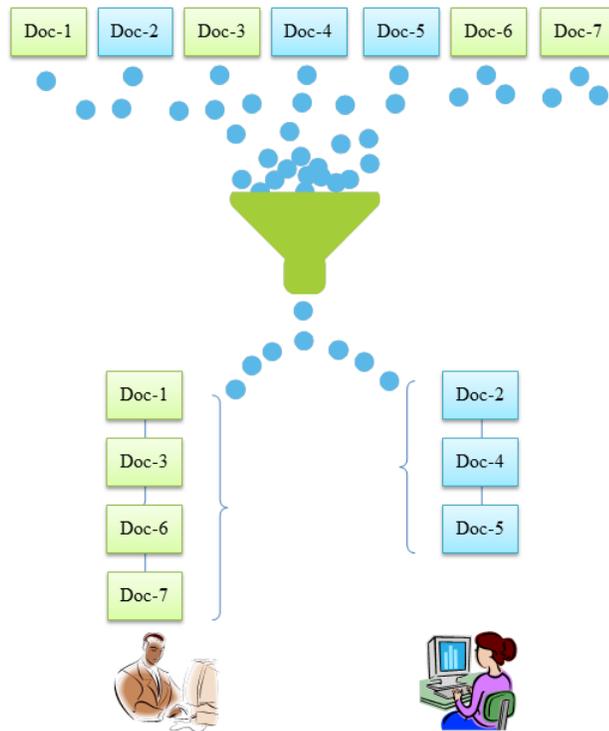


Figure 13 Information Filtering

Personalized retrieval can be achieved by enriching the user's query and filtering and ranking the retrieved results. Below we discuss these topics.

2.3.1 Query Enrichment

Users might not be able to use representative keywords to search and locate the desired information (38). As a result, *query enrichment*, also called *query expansion* and *query*

adaptation, enriches the user's query with extra keywords to retrieve relevant results (80). Query enrichment implicates the weight and significance of the keywords. There are different techniques for query enrichment, as classified in (38) into six categories based on two factors: whether the enrichment is based on the user (user-focused) and whether the enrichment is implicit or explicit. The six categories are listed below.

The first category is query enrichment based on processing the user's profile implicitly: selecting the expansion keywords from the user's profile. An example is in (55), where the process starts by identifying the related documents from the user's profile repository that contain similar query keywords. Then the documents are re-ranked based on these keywords using weighting schema for modified term frequency (TF). Top documents are then selected, and their keywords are sorted based on document frequency (DF) weighting schema. Finally, the top four keywords are selected as expansion keywords and used in the query enrichment. Another example is in (81), where the user uses defined tags, and these tags are used in the query expansion. A statistical model for tags is created and used to identify the relevant keywords from the user's profile. Other examples of systems and applications that use this technique can be found in (49) and (57).

The second category is query enrichment based on implicit pseudo-relevancy feedback: selecting the expansion keywords from top relevant results and their snippet. This involves the full cycle of retrieving the results of the user's query and then expanding the query by selecting the keywords from top relevant results. Examples of systems and applications that use this technique can be found in (82), (83), (84), and (85). The author of (86) mentioned two disadvantages of the pseudo-relevancy feedback technique. First, it adds overhead to the system by performing two search rounds for each query to get the expansion keywords first and then to retrieve the relevant documents to the expanded

query. Second, the first cycle brings the expansion keywords and depends on the assumption that the returned documents are relevant, which is not always guaranteed.

The third category is query enrichment based on processing the user's usage logs implicitly: selecting the expansion keyword from the usage logs including previous queries and the visited result. An example is in (86), where the author expanded the query by using a machine learning technique to identify the similarities between the user's logs and the queries. The author addressed the two disadvantages of pseudo-relevancy feedback by using a snippet of the document to know how relevant the retrieved document was and limited the selection to the snippets that exceeded a certain threshold to assure the relevancy. Other examples of systems and applications that use this technique can be found in (44), (87), and (88).

The fourth category is query enrichment based on implicit global analysis: selecting the expansion keywords from a thesaurus (e.g., WordNet, a source of knowledge, Wikipedia, or a big corpus based on co-occurrence statistics). The user's query is expanded with other semantically related keywords. TF-IDF is used frequently in this technique to determine the weight and importance of the expansion keywords along with a defined threshold that limits the selection to the top relevant keywords. Global analysis is not user focused because the user is not involved in the query-enrichment process. Examples of systems and applications that use this technique can be found in (89), (90), (91), and (92).

The fifth category is query enrichment based on explicit relevancy feedback: selecting the expansion keywords based on the explicit feedback provided by the user on the retrieved result relevancy. The user is asked to provide feedback on the returned result. The feedback can be either positive or negative, which corresponds to relevant or irrelevant results. The expanded query is then used to retrieve the positive rated results and filter out

the negative rated results. Examples of systems and applications that use this technique can be found in (93) and (94).

The sixth category is query enrichment based on explicit interactions with the user: selecting the expansion keywords based on interactions with the user by showing the user a candidate set of expansion keywords suggested by the system and then having the user select the appropriate one. An important step in this technique is that the system first produces a ranked list of keywords to be re-evaluated by the user where these keywords are related to the user's query. This technique emphasizes users and their role to expand the query. Examples of systems and applications that use this technique can be found in (95), (96), and (97). In (96), the author mentioned that the interaction technique is more efficient than other automatic techniques, but it depends on presented user interface and human judgment, which varies based on the user knowledge of the domain.

Both techniques, implicit pseudo-relevancy feedback and global analysis, are not user focused, as they do not depend on any user feedback or profile. The other four techniques are user focused, as they depend on some sources of user information.

2.3.2 Results Filtering

Filtering results is a major milestone in retrieving personalized results. It helps in returning only the results relevant to the user's needs and filtering out the irrelevant ones. Different techniques are used in results filtering.

In (98), the filtering techniques were classified into three categories. The first is content-based filtering, which involves analyzing each item to assess what is interesting to the user based on the user's profile (99) (100) (101) (102). The second is collaborative-based filtering, which involves collecting opinions from people to direct the user to similar

opinions (103) (104) (105) (106) (107) (108) (109) (110) (111) (112) (113) (114). The third is the hybrid approach, which combines the content-based and collaborative-based approaches (115) (116) (117) (118).

Another classification is found in (59), where the authors have similar classifications to the one in (98), but they further classified the collaborative-based filtering into two categories: traditional collaborative-based filtering and model-based collaborative filtering. The model-based collaborative filtering is classified further into four categories. The first is item-based collaborative filtering, which involves an offline process to build an item similarity matrix based on the item information (103) (104) (105). The second is cluster-based collaborative filtering in either user-based clustering or item-based clustering (106) (107) (108). The third category is association and sequence rule-based approaches, which discover patterns for the association and sequence of items (109) (110) (111) (112). The fourth is graph theoretic approaches, which transform collaborative filtering rating data into a directed graph where users are represented as nodes and edges represent the predicted users (113) (114). In addition, (59) looks at different aspects to classify these techniques, such as individual versus collaborative, reactive versus proactive, user versus item information, memory based versus model based, and client side versus server side.

2.3.3 Results Ranking

Most of ranking functions used in web search results are trained using machine learning algorithms. This training is done either through collecting explicit feedback from the users judging the relevancy of specific results or through implicit feedback by analyzing users'

clicks and click-through data. Thus, the search results improvement is generic and not specific to a certain user who has some interests (65).

In (119), the authors proposed a new technique that improves the semantic search by assigning a weight to different semantic relationships. In addition, the number of meaningful relationships between resources and keywords is taken into account as well as the coverage of the keywords. The use of this information results in getting more accurate results to the user. The proposed technique has been tested with real-world data and found to be more accurate than previous ranking models.

In (120), the authors used a naming authority to connect an identifier (URI) to the source that has the authority to assign that identifier. The notion of naming authority can be generalized to other identifier schemes to establish a connection to the provenance of the identifier, such as a person or an organization. The authors derive a naming authority matrix from a given dataset and use the PageRank algorithm to determine rankings for sources. After that, another algorithm is used to rank individual identifiers based on the values assigned by their sources. The proposed method is schema independent, requires no manual input, and has applications in search, query processing, reasoning, and user interfaces over integrated datasets. This work demonstrates a set of scalable algorithms for ranking over a general model of structured data collected from an open, distributed environment based on the notion of naming authority. The authors adapted the general model to accept RDF and import the intricacies of RDF data from the web. In comparison to plain PageRank on a node-link graph representation of RDF, the proposed methods exhibit similar runtime properties while improving the quality of the calculated rankings. Compared to other methods that require manual input by a domain expert to specify schema weights, the proposed method derives rankings for all identifiers in the dataset

automatically. The methods were tested on real-world web datasets that contained 1.1 billion data items from 6.5 million web sources. The experiment provides evidence for improving the quality of the rankings with a user study of 36 participants.

In (121), the authors proposed a unique architecture for a personalized semantic search engine (PSSE). A PSSE is a crawler-based search engine that enables multi-crawlers to collect resources from both semantic and traditional web resources. The system learns the users' interests and preferences automatically from the web usage data and uses these data to rank the results. The ranking and final score of the search result is calculated using the traditional link analysis, content analysis, and weighted user profile.

In (65), the authors proposed a new approach for personalizing the results of web searches for a specific user. The new model indicates the relevancy of documents for specific users when they provide a certain query. The model has an input, which is a compact user profile that will be used to generate user-specific search results. Users' profiles are captured and trained using their search history over a long period. This approach uses probabilistic models for predicting the relevancy of documents to specific users on a certain query. Using one discrete variable for each document to specify the topic of the document, there is a preprocessing step using a text-based classifier to identify the topic of each document. This is using human-generated ontology provided by the Open Directory Project (ODP, dmoz.org). This preprocessing step helps in calculating personalization ranking quicker when taking the user's query. In addition, for each user there is a variable stating the type of documents the user is looking for using the query and the user's history. The probabilistic model was experimented in (65) on with historical search data from thousands of users of the major search engine, Bing, by using the queries and search result clicks to build long-term user profiles for the users' interests.

This profile is used to calculate the relevancy of the query based on the user's history. The authors found some improvements in retrieval performance for queries with high ambiguity and major improvements for acronym queries. Although the proposed approach is simple as it is using topics to indicate relevancy, there is little computational overhead caused by the preprocessing step to calculate the topics for all documents and then recalculating the probability of relevancy for each user using the user's profile. This approach uses a compact user profile, which is a topic-based profile. The capturing and learning of the profiles are based on a user's long-term search history.

2.3.4 Conclusion

For query enrichment, we use a hybrid technique that combines the implicit processing of the user's profile with the global analysis based on the domain ontologies and explicit feedback. The motivation behind the hybrid technique is to overcome the limitations of these techniques by combining them and giving more weight to the user to imply the expanded query. Second, we integrated this with health and food domain ontologies to take advantage of their knowledgebase.

For filtering the results, we use the content-based filtering technique as our work is based on a defined ontology and knowledgebase for food and health. Furthermore, the user's profile is ontology based and associated with the domain ontology.

We combine different techniques to rank the results based on the user's profile and on related information from the domain ontologies and knowledgebase. Furthermore, we consider the frequency of the results between specific concepts. The higher frequency of a certain predicate in a different data sources indicates that this information is more trustable. Therefore, we give it more weight.

2.4 Agent-Based Framework for Health and Nutrition Information

In this section, we define agents and frameworks and then define the scope of the health and nutrition information on which we focus in our research. After that, we research work related to this area.

There are many definitions and descriptions of software agents in the literature. Many of these definitions are listed in (122), such as, “Agents are computational systems that inhabit some complex, dynamic environment, and sense and act autonomously to realize a set of goals or tasks.” We go with a simple and comprehensive definition of agent as a software entity that has certain objectives, works autonomously in a specific environment, and collaborates with other agents (123).

2.4.1 Scope of Health and Nutrition Information

Health is defined by the World Health Organization (WHO) as "a state of complete physical, mental, and social well-being and not merely the absence of disease or infirmity," as mentioned in the broader sense in 1964 (124). *Health care* or *medical care* as mentioned in Oxford English Dictionary⁴ is related to diagnosing, treating and preventing diseases. The human body consists of *body parts*, such as eyes, a nose, and ears, and these parts perform *body functions*, such as vision, smelling and hearing.

Food is any consumed, eaten, drunk, or injected element by the body that provides nutritional value, comes normally either from plant or from animal, and consists of a number of nutrients, such as minerals, fats, proteins, carbohydrates,⁵ and vitamins.⁵ The human body consumes food and produces energy to grow and maintain life.

⁴ <http://www.oxforddictionaries.com/>

⁵ <http://global.britannica.com/EBchecked/topic/212568/food>

Nutrition focuses on food's impact on the human body. By consuming the right food, we can avoid some health issues. The diet arranges what food to eat and its quantity to keep the body healthy. There are two types of health professionals: (1) dietitians or nutritionists (the common name) who deal with human nutrition and meal planning and (2) clinical nutritionists who deal with the effect of nutrition on clinical diseases and the relation between with nutrition and drugs (125).

In our research, we limit the health aspects on the relations between diseases, body parts, and body functions with food and its nutrients. We deal with nutritionists to validate the results of our research. We have selected the health and nutrition domains for our case study due to their importance, the demand of the users and experts in this field, and the limited research in this area.

2.4.2 Related Systems

In (126), the authors studied the major challenges in health information system and retrieval (HIS/HIR) queries. More challenges were addressed in assessing the sources and quality of the health information the users find and act on. They concluded that there is an urgent need to research theoretical and practical HIS/HIR from the consumers' point of view. The authors stated that health care systems are becoming more patient centric, and consumers are controlling their own personal health choices. Finally, the authors recommended having a mechanism for joint efforts between consumers, providers, and decision makers to help achieve personalized health care. The authors did not consider any related cultural or lingual aspects of the user, and this personalization needs further detailed research.

2.4.2.1 HealthFinland

In (127), the authors present HealthFinland, which semantically publishes and retrieves health information. Its objective was to provide citizens with reliable, up-to-date, and relevant health information on the web by mixing resources from governmental, non-governmental, business, and other sources. It handles the user's point of view by addressing the challenge of finding content using basic vocabularies compared to technical medical terminology and the difficulties in retrieving relevant information from several sites. To resolve this, they developed an intelligent semantic portal for retrieving and presenting contents from health-interest perspectives. The limitation of this approach, as noted in the future work section, is to address the personalization based on user's profile information.

2.4.2.2 Personalized Health Information Retrieval System (PHIRS)

In (128), the authors addressed the challenge faced by consumers when seeking health information on the Internet. They proposed a personalized health information retrieval system (PHIRS) to recommend health information for consumers. The system consists of four modules:

- (1) The user-modeling module is responsible for getting the user's preference and related health interests.
- (2) The automatic quality-filtering module identifies the quality of the retrieved health information.
- (3) The automatic text difficulty-rating module helps classify the retrieved health information into two classes: professional or patient educational materials.

- (4) The user profile-matching module customizes the retrieved health information to match the individual's needs.

The authors conducted an initial test and showed that the results can assist health information consumers with a simple search that retrieves relevant information. The authors conclude that the initial test result shows that the evaluated pattern of semantic features in professional and consumer health is not enough. They suggest combining some surface features, such as structure, tense, and voice, with the used pattern and semantic features to help identify the text difficulty of health information, i.e. the use of technical and medical terms. Therefore, the limitation of this work is not having enough features to help identify relevant health information as well as not having sufficient testing for the proposed solution. The personalization here did not touch on the culture or language of the user.

2.4.2.3 CarePlan

In (129), the authors presented a new system, CarePlan, which generates customized patient-specific health care plans automatically. To determine the best clinical care plan, they utilized (1) the patient's medical personal profile, (2) up-to-date medical knowledge, (3) clinical pathways that are institution-specific, and (4) a personalized educational health care program. They came up with a new Semantic Web framework that allows for the synthesis of heterogeneous operational and medical information and knowledge resources and renders the technical basis for a services-oriented architecture to generate and orchestrate patient-specific health care plans. The authors concluded with sharing their belief that the Semantic Web will be the future way to get intensive knowledge and validate health care decisions, though it will face many challenges. The limitation of this

approach is the lack of the full implementation details as well as the food and nutrition information related to the patient. The personalization in this approach focuses on educational health information and does not talk about the culture or language of the user.

2.4.2.4 NOESIS System

In (130), the authors proposed NOESIS, a new adaptive searching mechanism using innovative technologies to obtain, use, and manipulate medical information. The authors highlighted that medical knowledge is inherently complex and uncertain and that medical experts may provide different interpretations for symptoms because all of them also depend on a given context and most of the interpretations are established by statistical utilization. So it is necessary to capture a whole knowledge to understand and take care of patients with cardiovascular diseases adequately. For this reason, the authors proposed a system aiming at being a valuable instrument for cardiologic medical information retrieval from heterogeneous, distributed medical databases that mediate medical decisions regarding critical health conditions. The proposed adaptation features generate a personalized searching process for the users depending on the information stored in their personal profiles. The approach is lacking the use of the Semantic Web as it adds much into such complex heterogeneous data sets and helps with the inference of the relevant information to the user. The personalization search approach does not consider the culture or language of the user.

2.4.2.5 MedSIFTER Model

In (131), the authors highlighted the explosive growth in number of information sources and that users now can access a wide variety of health information from the web. However, information that may be potentially relevant to individual users remains highly

scattered, and users frequently have to dig and aggregate information from multiple sites. The authors introduced MedSIFTER, a proposed trusted model, as a one-stop-shop access point to personalized health and medical information. The model centralizes personal information management to facilitate specific information aggregation tasks of individual clients. It employs group query mixing and noise query mixing to hide users' profiles from external eavesdroppers. Experiments were conducted to demonstrate trade-off levels between retrieval performance and the degree of privacy preservation in the proposed query mixing strategies. This trade-off did not consider personalization from the user's culture or language point of view.

2.4.2.6 Cobot System

In (132), the authors described a mixed initiative socio-semantic conversational search and recommendation system for finding health information. They argued that users could do a live conversation about their health issues by using the proposed system. Then the collaboration mode would bring relevant users into the same conversation and provide context-based recommendations related to the conversation subject. The authors then illustrated the power of their search, which returns relevant search directly or via other users without using the conventional search engines, which they believed often confuse and frustrate users. The recommendation was based on the social context, not on personalization factors. Moreover, the personalized culture and language were not mentioned.

2.4.3 Criteria for Evaluation

We list below some criteria we have selected to compare the above systems.

2.4.3.1 Query Manipulation Criteria.

We have set the following criteria to evaluate the used techniques for query manipulation in the related systems:

- Query type: the accepted types of user's queries, such as question or semantic
- Query natural language scope: the scope of accepted natural language, such as free text and controlled text
- Query processing method: the way it processes the query, such as NLP
- Query templates usage: whether or not it uses any type of query templates

2.4.3.2 User's Profile Criteria.

We have set the following criteria to evaluate the used techniques for a user's profile in the related systems:

- Profile existence: whether or not it has a user's profile
- Culture preferences inclusion: whether or not it has a user's culture preference
- Profile representation: how it represents the user's profile
- Profile location: where it stores the profile

2.4.3.3 Personalized Retrieval Criteria.

We have set the following criteria to evaluate the used techniques for personalizing the retrieved results in the related systems:

- Query enrichment: whether or not it enriches the user's query
- Result filtering: the methodology used in filtering the results
- Result ranking: the way it ranks the results

2.4.3.4 Framework Criteria.

We have set the following criteria to evaluate the used techniques for the framework in the related systems:

- Agent-based: whether or not it is agent based
- Ontology support: whether or not it is semantic based and supports ontologies
- Domain dependent: whether it is domain dependent or is open for any domain
- Multilingual: whether or not it is multilingual

2.4.4 Comparison

In the following sections, we compare the reviewed systems with regard to the criteria we have defined in the previous section.

2.4.4.1 Query Manipulation Criteria

TABLE 1 shows the comparison results based on the query manipulation criteria.

TABLE 1 Comparing the Related Systems on the Query Manipulation Criteria

System/Criterion	Query type	Query natural language scope	Query processing methods	Query template usage
Health-Finland	Question	Controlled text	NLP	No
PHIRS	Question	Controlled text	String matching	No
CarePlan	Question	Free text	NLP	No
NOESIS	Question	Controlled text	String matching	No
MedSIFTER	Question	Controlled text	String matching	No
Cobot	Chatting	Controlled text	String matching	No

Our analysis of the comparison is that the current systems do limited query processing by using only one technique, either NLP or string matching. They do not use the query template for processing the questions or retrieving the answers. Therefore, with these limitations we are motivated to develop semantic query manipulation that utilizes multiple-query processing techniques and uses the query template in matching the user's query with the domain ontologies and knowledgebase to retrieve relevant answers to the user.

2.4.4.2 User's Profile Criteria.

TABLE 2 shows the comparison results based on the user's profile criteria.

TABLE 2 Comparing the Related Systems on the User's Profile Criteria

System/Criterion	Profile existence	Culture preference inclusion	Profile representation	Profile location
HealthFinland	No	N/A	N/A	N/A
PHIRS	Yes	No	Text	Server
CarePlan	Yes	No	XML	Server
NOESIS	Yes	No	XML	Server
MedSIFTER	Yes	No	Text	Server
Cobot	No	N/A	N/A	N/A

As shown in TABLE 2, we find that the culture aspect of the user is not addressed and the ontology is not used to represent the user profile. In addition, all of the compared systems store the profile on the server without addressing the privacy issues. These limitations

motivate us to develop a framework that addresses personalization with respect to the user’s culture and privacy using an ontology-based user’s profile such that a great deal of reasoning can be done easily.

2.4.4.3 Personalized Retrieval Criteria.

TABLE 3 shows the comparison results based on the personalized retrieval criteria.

TABLE 3 Comparing the Related Systems on the Personalized Retrieval Criteria

System/Criterion	Query enrichment	Result filtering	Result ranking
HealthFinland	No	N/A	Relevancy
PHIRS	No	Content based	Relevancy and profile
CarePlan	No	Content based	Relevancy
NOESIS	No	Content based	Relevancy and profile
MedSIFTER	No	Content based	Relevancy and profile
Cobot	No	Collaborate based	Relevancy

Our conclusion from the comparison above is that the query enrichment is not used in the related systems. The results ranking techniques used in these systems do not consider all aspects of the user’s profile, and the used result filtering techniques are limited. These limitations motivate us to invest in query enrichment, as it is a major milestone in personalization, filter the results using all aspects of the profile, and rank the results with respect to the relevancy by giving more weight to user feedback.

2.4.4.4 Framework Criteria.

TABLE 4 shows the comparison results based on the predefined criteria for the framework.

TABLE 4 Comparing the Related Systems on the Framework Criteria

System/Criterion	Agent-based	Ontology support	Domain dependent	Multilingual
HealthFinland	No	Yes	Yes	Yes
PHIRS	No	No	Yes	No
CarePlan	No	Yes	Yes	No
NOESIS	No	No	Yes	No
MedSIFTER	No	No	Yes	No
Cobot	No	Yes	Yes	No

Our conclusion from the above comparison is that the evaluated systems are not agent based but are domain dependent, limiting their scalability. Moreover, the majority of the evaluated systems are monolingual, and thus they will not be useful for the users of other languages. We are heavily motivated to develop an agent-based multilingual framework that can fit any domain and help in retrieving relevant results based on semantic query manipulation and personalized retrieval.

2.4.5 Conclusion

Based on this survey, there is lack of cultural- and lingual-based personalization for the health, food, and nutrition domains that will help in giving better recommendations to users. Hence, we extend the current approaches by building a framework for a cross-

cultural and cross-lingual multi-agent recommendation tool having an ontology-based user's profile to retrieve relevant health and nutrition information.

CHAPTER 3

FRAMEWORK FOR ONTOLOGY-BASED SEMANTIC ANNOTATION AND PERSONALIZED INFORMATION

RETRIEVAL (OSAPIR)

This chapter introduces the main framework for ontology-based semantic annotation and personalized information retrieval (OSAPIR), where this thesis focuses on the semantic query manipulation and personalization component of OSAPIR. This thesis research work is a part of project No.10-INF1381-04 supported by King Abdulaziz City for Science and Technology (KACST) through the Science & Technology Unit at King Fahd University of Petroleum & Minerals under the National Science, Technology and Innovation Plan (NSTIP). The aim of the project is to build a semantic retrieval portal for health and nutrition information.

3.1 Introduction

Web content is growing exponentially, which brings many challenges in accessing the information. Meanwhile, users' demands to find relevant information have increased. Most people use the traditional search engines to locate information, such as Bing, Google, and Yahoo. Not all users are satisfied with the current search engines because they do not find the search results relevant to their needs. This is obvious when they

search for critical information, such as health and nutrition, where they desire more relevant and precise information than they can get through traditional search engines selected from trusted sources.

3.1.1 Multilingual Web Content

Although most web content is presented in English (56%), there is still a great amount of content in other languages.⁶ Traditional web access to cross-lingual content is only possible if websites are translated into the desired language. There is a lack of explicit mechanisms to reconcile automatically information expressed in different languages. This leads to situations in which data expressed in a certain language are not easily accessible to speakers of other languages. The Semantic Web offers a great opportunity to make web information broadly accessible, independent of culture and native language.

3.1.2 Cross Domains

Many different knowledge experts are working in their area of expertise not only independently, but in isolation. Such nonintegrated knowledge, when searched with current search engines, can answer users' questions with no relation or semantic understanding between domains. The Semantic Web can play a very important role by providing understanding and the semantics of a given domain. We are motivated by the requirement of semantically integrating the knowledge from heterogeneous domains. This cross-domain integrated knowledge should enable us to answer users' questions referring to multiple domains by semantically understanding the query and reasoning the answer based on the relations among the domains.

⁶ <http://www.netz-tipp.de/sprachen.html>

3.1.3 Relevancy

Search engines crawl web content and create indices that are used to retrieve the results for users' search queries. The users write their queries using natural language, while the current search engines are keyword-based. This leads to a challenge to understand users' queries correctly. Moreover, users might not be able to express all their needs explicitly while the search engines are limited to the provided query to bring the matched results. Because user's needs are different, the relevancy of the retrieved results varies from user to user. This leads to a challenge to get the relevant and personalized information based on the user's needs. The Semantic Web addresses relevancy by semantic understanding of the users' queries and reasoning on the annotated web sources based on the integrated domain ontologies. Moreover, personalization technologies help in understanding the users' needs better, which can support semantically enriching the queries and retrieving personalized results. This raises the challenges of semantically manipulating the users' queries, reasoning, and annotating web content based on the domain ontologies.

3.1.4 Framework

Some domains are quite critical to users, such as the health and food domains. Information retrieval in these areas makes these challenges even more obvious. There is a need to have an integrated infrastructure that handles these challenges. An infrastructure in the form of a framework with support of the Semantic Web and personalization technologies will help the web developers to develop semantic applications for different domains.

A framework is a software platform for developing an application for a given platform. Generally, frameworks provide an application programmable interface (API) for

accessing its components, whereas the framework itself serves as pillars for building up the application so developers do not have to do everything from scratch. A framework may also include additional software libraries and other programs used in the software development process. Therefore, these are considered basic requirements for any common framework for development.

We propose a framework for ontology-based semantic annotation to retrieve personalized information (OSAPI). Below, we present the proposed framework to handle multilingual cross-domain web content and that can be easily adapted to any domain, such as the health and food domains. We start with discussing the requirements of such a framework, show the proposed framework architecture, and briefly describe each component of the framework.

3.2 Requirements

We aim to build a multilingual cross-domain personalized Semantic Web search framework that can adapt to any domain, such as the health and food domains. Below we present the requirements for such a Semantic Web search framework.

- 1) The framework should be applicable to any domain with minimal customization.
- 2) The framework should support multilingual needs with respect to ontologies, Web sources, knowledgebases, and user's queries.
- 3) The framework should facilitate cross-domain integration of ontologies and knowledgebases.
- 4) The framework should support acquiring and annotating web sources in heterogonous formats.

- 5) The framework should provide a mechanism to decide the trust level of the acquired web sources.
- 6) The framework should generate standard semantic annotation formats for the acquired web sources based on the domain ontologies.
- 7) The framework should semantically manipulate the user's queries.
- 8) The framework should provide reasoning capabilities for answering user's queries.
- 9) The framework should capture and model the user's preferences.
- 10) The framework should personalize the retrieved results.
- 11) The framework should support a standard ontology representation format.
- 12) The framework should provide the required ontology management services to achieve the desired objectives (i.e., alignment of ontologies from different domains and languages).

3.3 Proposed Framework

Based on an intensive literature review and discussions among the project team members including the consultants, we propose an ontology-based semantic annotation and personalized information retrieval (OSAPIR) framework that addresses the above requirements. The proposed framework is able to adapt to any domain by defining the domain ontologies, lexical resources, trust level, and seed web sources. Furthermore, the framework supports multilingual needs regarding ontologies, web sources, and users' queries. Figure 14 shows the architecture of the proposed OSAPIR framework.

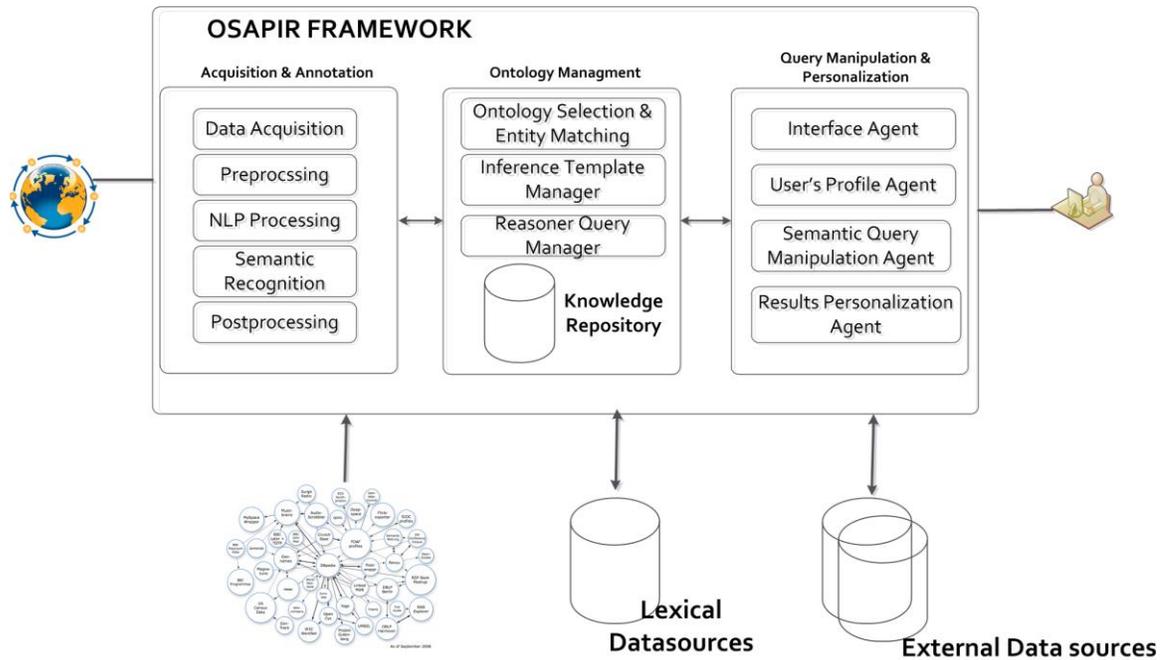


Figure 14 Architecture of OSAPIR Framework

There are three dimensions of the requirements that work together to achieve the framework's objectives. First, users' queries need to be semantically understood according to the domain ontologies. The retrieved results from the knowledgebase should be personalized based on users' needs. Second, the web content needs to be annotated according to the domain ontologies to populate the knowledgebase. Third, the cross-domain ontologies and knowledgebase need to be managed in an efficient and effective way. As a result, the proposed framework is divided into three major components: data acquisition and semantic annotation, ontology management, and semantic query manipulation and personalization. Below is a brief description of each component.

3.3.1 Data Acquisition and Semantic Annotation Component

The main goal for this component is to collect and annotate the contents of multilingual web sources based on the predefined domain ontologies. This component consists of two major layers: the acquisition layer and the semantic annotation layer.

The acquisition layer consists of multiple data integration tasks for collecting data from web sources related to the targeted domains. The data collected from web sources are then used by the annotation layer for semantic enrichment. The acquisition layer is configured to collect data from specific websites based on certain criteria such as trust level or predefined seed websites. The relevant web sources are collected based on their relevancy to the domain ontologies. This layer supports processing of all common web document formats such as HTML, XML, PDF, Office Word, and multimedia.

The semantic annotation layer annotates the acquired web sources based on the domain ontologies and the predefined cross-domain integration. Moreover, it provides multiple mechanisms to perform automated annotations for semi-structured (i.e., tables) and unstructured (i.e., paragraphs) web sources. This layer produces embedded annotation inside the web document using standard annotation languages such as RDFa, Microformat, and Microdata. It can also produce stand-alone annotation using standard annotation languages such as RDF, N3, and Turtle. More elaboration about this component can be found in (133).

3.3.2 Ontology Management Component

The ontology management component takes care of managing the network of heterogenous ontologies and knowledgebases required by the OSAPIR framework (i.e., integration model for cross-domain and/or multilingual ontologies). It also provides

different ontologies management tasks for information processing (i.e., mapping of various ontologies for more efficient sharing and reuse). This component can process any standard ontology representation language. It also provides API interfaces to access the ontologies by two other components of the proposed framework and provides reasoning capabilities on the knowledgebase to allow semantic answering of the users' queries. More elaboration about this component can be found in (134).

3.3.3 Semantic Query Manipulation and Personalization Component

This component is used to interface with the end user and capture and model the user's preferences into a user's profile. It semantically manipulates the multilingual user's queries and enriches them with more information from the user's profile. This component interacts with the ontology management component for query reasoning based on the domain ontologies and knowledgebase. Moreover, it personalizes the retrieved results and captures the user's interactions to enhance the user's profile and provide answers that are more relevant. This component is the main focus of this thesis, and more elaboration about the details of this component will be highlighted in the upcoming chapters.

CHAPTER 4

AGENT-BASED-FRAMEWORK FOR SEMANTIC-QUERY-

MANIPULATION AND PERSONALIZED INFORMATION

RETRIEVAL (ASPIR)

The proposed framework in this chapter represents one component of the OSAPIR framework introduced in Chapter 3, which aims to help users find relevant information that fits their needs. The architecture of the main framework is composed of three major components. The first is the semantic query manipulation and personalization component, the focus of this thesis, which takes care of representing the user's preferences, understanding the user's queries semantically, and personalizing the retrieved information. The second is the ontology management component, which takes care of representing and managing the domain ontologies. The third is the data acquisition and semantic annotation component, which takes care of determining trusted web sources and annotating the information based on the predefined domain ontologies. In the following sections, we highlight the main requirements for the semantic query manipulation and personalization framework, namely the agent-based-framework for semantic-query-manipulation and personalized information retrieval (ASPIR). Then we show the architecture of the ASPIR framework followed by detailed explanations of each agent in the ASPIR framework.

4.1 Framework Requirements

Usually, we need a framework to abstract the functionalities of a system. The main requirements of a semantic query manipulation and personalized information retrieval framework are that it should:

- 1) capture and model the user's preferences;
- 2) semantically manipulate the user's query;
- 3) enrich the query with the user's preferences;
- 4) personalize the retrieved results;
- 5) support multilingual use;
- 6) be domain independent; and
- 7) support building a friendly interface.

4.2 Framework Architecture

We propose a framework for the semantic query manipulation and personalized information retrieval system that meets the requirements mentioned in Section 4.1. The proposed framework is agent based, whereby the agent helps in adapting to the user's needs and learns the user's preferences (135). An *agent* is a software entity that has certain objectives, works autonomously in a specific environment, and collaborates with other agents (123). As there is much agent interaction, using agents is advised to utilize the well-established and efficient agent communication mechanisms that ease communication complexities (136). Moreover, agent-based modeling adds to the information retrieval system the following three advantages (137):

- (1) Adaptability: agent can monitor the user's behavior to learn the user's preferences, to understand the user's needs, and to update the user's profile.
- (2) Initiative: agent can proactively return the relevant information depending on the user's needs and observe any variation in the information sources.
- (3) Collaborative: agents can collaborate with each other to share the information.

The proposed framework consists of: (1) the interface agent, which handles the user's interactions, (2) user's profile agent, which captures and manages the user's preferences, (3) semantic query manipulation agent, which manipulates the user's query, and (4) personalized retrieval agent, which personalizes the retrieved results. Figure 15 shows the ASPIR architecture followed by the details of each agent in the framework.

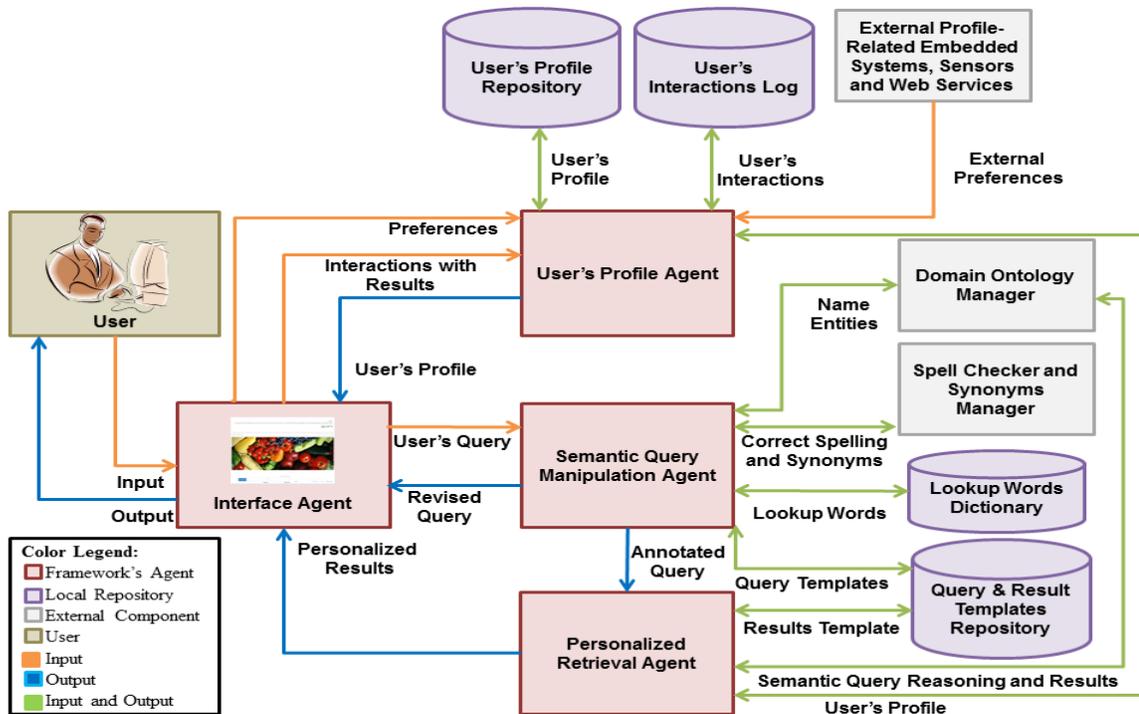


Figure 15 Architecture of the ASPIR Framework

4.3 Interface Agent

The interface agent is the endpoint from the user's perspective that allows the user to create a profile. It accepts the user's queries, formulates and shows the personalized results, and then monitors the user's explicit and implicit behaviors that reflect the user's feedback on the results.

The interface agent needs to interact with the user to get the user's input and display the retrieved results. The input could be the preferences entered explicitly by the user, which will be forwarded to the user's profile agent to update the user's profile. The input could be also the user's queries, which will be forwarded to the semantic query manipulation agent for semantic manipulation. The semantic query manipulation agent will communicate back to the user if there is a need to revise the user's queries, get any missing information, or correct the spelling. Moreover, the interface agent displays the user's profile and formulates the personalized results to the user. Finally, it monitors the user's interactions on the retrieved results and forwards these interactions to the user's profile agent to infer new preferences. Figure 16 shows the functionalities of the interface agent followed by more emphasis on three major functionalities, namely:

- 1) collecting the user's preferences;
- 2) monitoring the user's behaviors; and
- 3) formulating the personalized results.

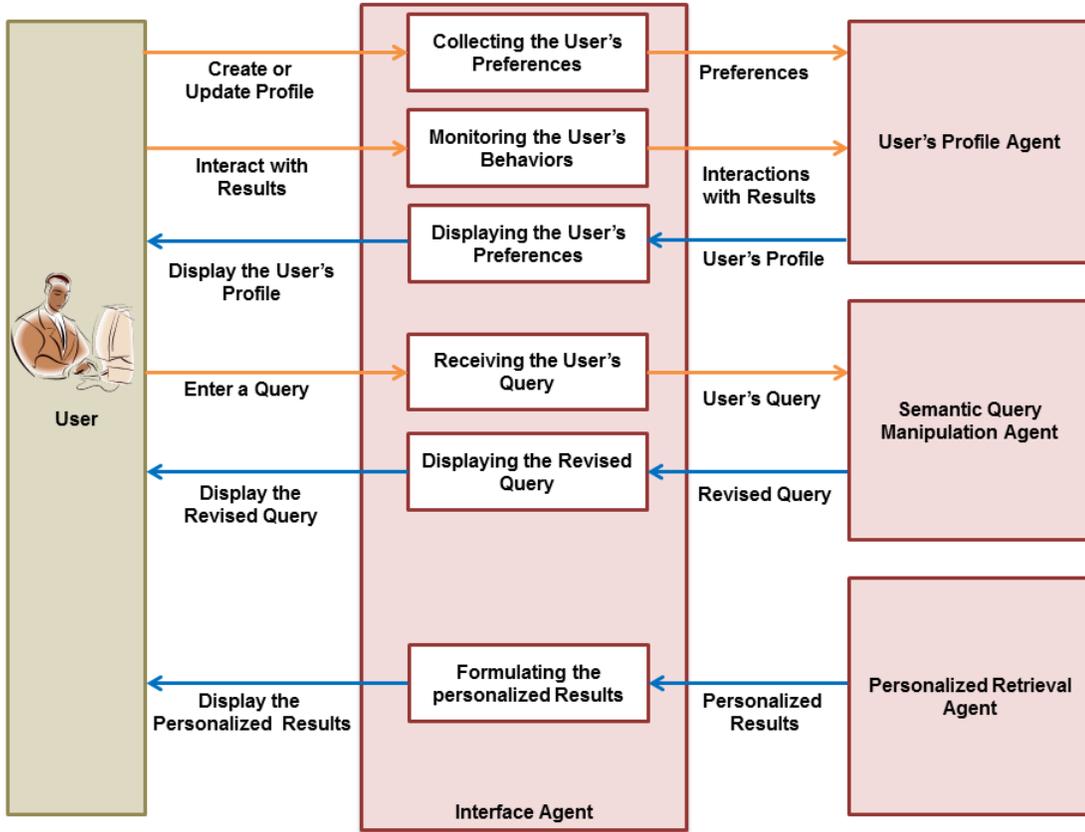


Figure 16 Interface Agent Functions

4.3.1 Collecting the User's Preferences

One of the major functionalities of the interface agent is to collect the user's preferences by asking the user explicitly to fill out a form. The preferences are defined based on a methodology, as will be explained in Chapter 5. The preferences are then sent to the user's profile agent to create a new profile or to update an existing one. The user can always access the profile's form and update it. Moreover, the user can prioritize the preferences to give more weight to the important ones from the user's perspective. We will give more details about the prioritizing in Section 4.4. Given a user u_m and set of X preference elements $pe_{m1}, pe_{m2}, \dots, pe_{mx} \in PE_m$ where the user can define a value for each preference $vpe_{m1}, vpe_{m2}, \dots, vpe_{mx}$ and associate a weight for each preference $wpe_{m1},$

$wpe_{m2}, \dots, wpe_{mx}$, the value of the preference depends on the nature of the preference, such as a “milk” value for an “allergy” preference. The value of the preference’s weight varies from 0 to 1 based on the user’s assessment of how a certain preference can influence the food choice. The possible options for these weights are: very important ($w=1.00$), important ($w=0.75$), neutral ($w=0.50$), not important ($w=0.25$), and not applicable ($w=0.00$). The weight of each preference is based on the user’s inputs and used in the results filtering as discussed in Chapter 5. It shows that the initial weights are based on a survey conducted to prioritize the results.

4.3.2 Monitoring the User’s Behaviors

Another major functionality of the interface agent is to monitor the user’s behaviors and interactions on the retrieved results. To capture their preferences, users can fill out a form to create a profile for their preferences, but most users do not spend the time to fill out such forms (51). The interactions with the retrieved results will be logged and then analyzed to deduce new preferences and update the user’s profile. This helps in enriching the user’s queries and better personalizing the retrieved results.

The user’s interactions with the results can be either explicit or implicit. The explicit interactions are captured by asking users about their feedback on the results. Given a user u_m who types a query Q_I that has been processed and results Y are retrieved, the results consist of n number of predicates $P_1, P_2, \dots, P_n \in P$. Each predicate in the results comes from a certain web source D_1, D_2, \dots, D_n . Also, each predicate contains a number of RDF terms RT_1, RT_2, \dots, RT_p . RDF terms are part of the knowledgebase and are domain dependent. For example, in the food and health domains, “apple” is an RDF term found in the domain knowledgebase. We use Boolean functions to represent the captured explicit

feedback, where a Boolean function has two possibilities: true or false. The explicit feedback measures are listed below.

- For measuring the overall satisfaction of the results, an example question is, “Are you satisfied with the retrieved results?” This measure is represented by a Boolean value of the function: *satisfiedResult*(U_l, Q_l, P). The value of the function will be reflected in future searches.
- For identifying the predicates that should be filtered out from the results, an example question is, “Which predicate should be filtered out?” This measure is represented by a Boolean value of the function: *filterPredicate*(U_l, Q_l, P_x) to filter out the predicate R_x . For example, the result contains five predicates. Four of them are satisfied, while one is not satisfied. Then the user marks the unsatisfied predicate to be filtered out.
- For identifying if a certain predicate should be re-ordered and displayed first in the results, an example question is, “Which predicate should be shown first?” This measure is represented by a Boolean value of the function: *showPredicateFirst*(U_l, Q_l, P_x) for the predicate P_x . For example, the result contains eight predicates, which are ordered and sorted. The user thinks that the fifth predicate should be shown first. The user then marks the fifth predicate to be shown first for similar queries.
- For identifying the data sources that the user trusts more, an example question is, “Which data source do you trust more?” This measure is represented by a Boolean value of the function: *trustSourceMore*(U_l, Q_l, D_y) for the data source D_y . For example, the result contains a number of predicates that come from three different

sources. The user believes that a particular source is more trusted than the other two sources. The user marks this source.

- For identifying the RDF terms that the user likes and that could be added to the query for enrichment, an example question is, “What terms within the result should be added to the query?” This measure is represented by a Boolean value of the function: $likesTerm(U_1, Q_1, RT_z)$ for the RDF term RT_z of the result. For example, there are different terms within the results such as “apple” and “diabetes” in the statement, “An apple is good for diabetes.” The user believes that the term “apple” should be added to the query. The user then marks “apple.”
- For identifying the RDF terms that should be taken out from the query for enrichment, an example question is, “What terms within the result should be taken out from the query?” This measure is represented by a Boolean value of the function: $dislikesTerm(U_1, Q_1, RT_z)$ for the RDF term RT_z of the result. For example, there are different terms within the results such as “sugar” and “diabetes” in the statement, “People with diabetes should be careful when taking sugar.” The user believes that the term “diabetes” should be taken out from the query. The user then marks “diabetes.”

Meanwhile, the user’s implicit interactions can be captured by monitoring the user’s behaviors regarding the results. Given a user U_1 who types a query Q_1 that has m terms $QT_1, QT_2, \dots, QT_m \in Q_1$, the query is processed and then the results P are retrieved. The results consist of n number of predicates $P_1, P_2, \dots, P_n \in P$. Each predicate comes from a certain data source D_1, D_2, \dots, D_n . Also, each predicate contains a number of RDF terms RT_1, RT_2, \dots, RT_p . The user can click on a particular result and visit it staying for a certain time in the visited predicate. The visit duration is denoted as VD_1 . We use numeral,

Boolean. and array functions to represent the implicit measure. A Boolean function has two possibilities: true or false, the numeral function has number as value, while the array function is taking an array of inputs. The implicit measures are listed with examples.

- Logging the query terms to identify the frequently asked query's terms is represented by an array function $queryTerms(U_1, Q_1, QT)$ to log the terms of the query.
- Logging the clicks on a particular predicate to measure the visit frequency of the visited predicate is represented by a Boolean function $resultClicked(U_1, Q_1, P_x)$ to log the clicks on a particular predicate P_x .
- Logging the time spent in visiting a particular predicate to measure the possibility of preferring the visited predicate is represented by a numeral function $resultVisitDuration(U_1, Q_1, P_x, VD_1)$ to log the time spent VD_1 in visiting the predicate P_x .
- Logging whether the user prints a particular predicate to measure the possibility of preferring the printed predicate is represented by a Boolean function $resultPrinted(U_1, Q_1, P_x)$ to log the printing of the predicate P_x .
- Logging whether the user bookmarks a particular predicate to measure the possibility of preferring the bookmarked predicate is represented by a Boolean function $resultBookmarked(U_1, Q_1, P_x)$ to log the bookmarking of the predicate P_x .
- Rating a particular predicate to measure the possibility of preferring the rated predicate is represented by a numeral function $resultRated(U_1, Q_1, P_x)$ to log the rating of the predicate P_x . We use star to represent the rating where we have the following interpretations for the star ratings shown in TABLE 5.

TABLE 5 Star Ratings Details

Star rating	Weight	Details
5 stars	1.00	Very interested
4 stars	0.75	Interested
3 stars	0.50	Neutral
2 stars	0.25	Mildly interested
1 star	0.0	Not very interested

All explicit and implicit feedback is collected and sent to the user's profile agent for further processing to infer new preferences, to update the user's profile, and then to personalize the results. More details on how we use these measures are in Section 4.4

4.3.3 Formulating the Personalized Results

Another major functionality of the interface agent is to formulate the retrieved results. It formulates the semantic results that are retrieved from the personalized retrieval agent to show the results in a user-friendly interface. It shows the personalized results with the associated explicit feedback controls that are explained in Section 4.3.2.

We show the results in small boxes called semantic widget boxes. These include a set of results within the same relation based on a certain semantic query. The advantage of having such a way of representing the results is the flexibility of showing any set of results in any location on the screen. Another advantage of using the semantic widget box

is the ability to reuse these boxes in different result screens based on the user's question. This helps in avoiding extra effort to rebuild these boxes.

The content of the semantic widget box can be general information that fits any domain. For example, the user's profile widget box shows the profile information related to the retrieved results. Another example of general boxes is the manipulated query widget box, which shows how the query is manipulated and annotated. An example of the domain-dependent widget box is the positive relation widget box, which shows the results with positive relations between two different terms, such as "good food" for "diabetes." A final example is the abstract answer widget box, which summarizes the whole results. For example, if the user asks, "Is apple good for diabetes?" then the abstract answer widget box could show, "Yes, apple is good for diabetes based on 4 sources as shown in details below."

The expected result of any query is represented in a template. The results template is associated with semantic queries and contains three ordered lists of semantic widget boxes: the left semantic widget boxes list, center semantic widget boxes list, and right semantic widget boxes list. Each list can contain as much as needed of the semantic widget boxes based on the matched results template. Within each list, the semantic widget boxes are ordered so that they are shown in the result page with the same order. The result page is divided into three columns, left, center, and right, where each column is associated with the corresponding list: left semantic widget boxes list, center semantic widget boxes list, and right semantic widget boxes list. Figure 17 illustrates the results screen and shows the distribution of the semantic widget boxes. Given the result template RT that contains three lists of semantic widget boxes SWB_LEFT, SWB_CENTER, and SWB_RIGHT \in RT represented as following:

SWB₁-L, SWB₂-L, SWB_m-L for m semantic widget boxes \in SWB_LEFT

SWB₁-C, SWB₂-C, SWB_m-C for n semantic widget boxes \in SWB_CENTER

SWB₁-R, SWB₂-R, SWB_o-R for o semantic widget boxes \in SWB_RIGHT

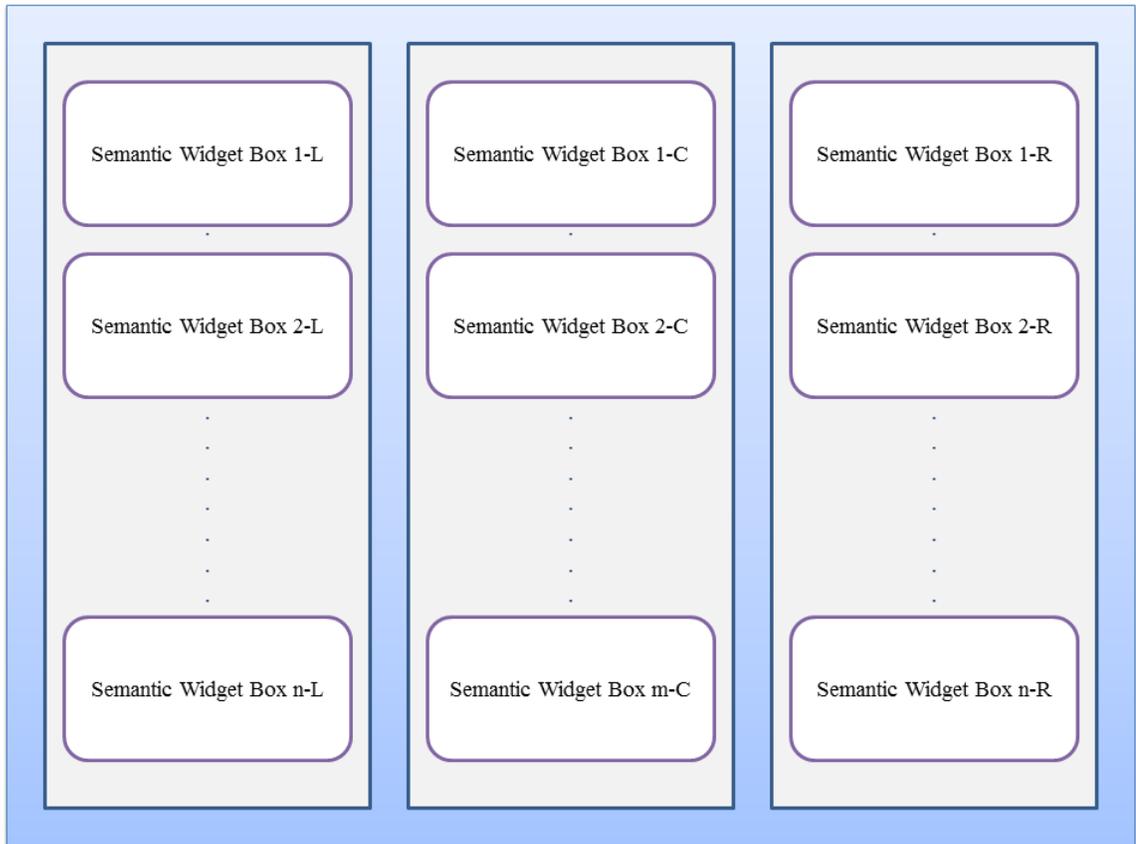


Figure 17 Distribution of Semantic Widget Boxes in the Results Page

The widget boxes can further shifted by the user to be more flexible with the user's preferences. The results template is then updated with the desired order of the results based on the user's interests.

4.4 User's Profile Agent

The user's profile agent manages the user's profile. More details on modeling the user's preferences and representing the user's profile are presented in Chapter 5 and Section 7.5. It logs the user's interaction with the results and then infers new preferences and updates the user's profile. It also helps the semantic query manipulation and enrichment agent to enrich the user's queries with more information from the user's preferences. In addition, it helps the personalized retrieval agent to personalize the results with information from the user's profile. The user's profile agent can also get feeds from external profile-related embedded systems, sensors, and web services. These functions are shown in Figure 18.

One major function of the user's profile agent is to learn and infer new preferences based on the user's interactions and behaviors. The preferences can be learned by analyzing the user's interactions log, which contains the user's interactions. The logs functions are summarized in TABLE 6 and are based on the functions mentioned in Section 4.3.2.

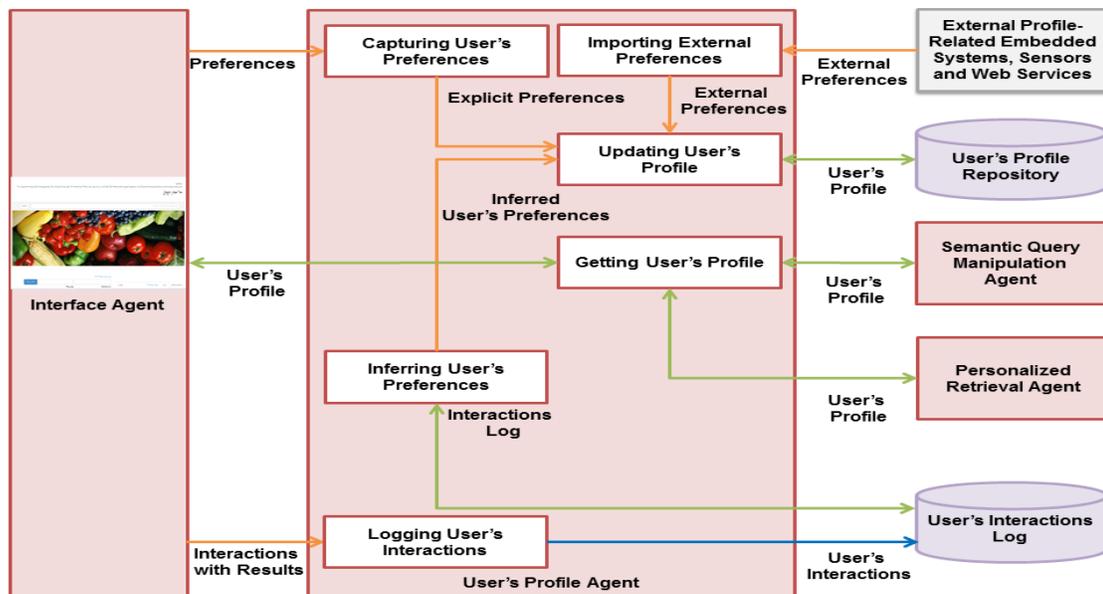


Figure 18 User's Profile Agent Functions

TABLE 6 User's Interactions Log Functions

Function	Description
$QueryTermFrequency(U_1, QT_1)$	Getting the frequency of a term QT_1 in the user U_1 queries
$RDFTermLikeScore(U_1, RT_1)$	Getting the like score of an RDF term RT_1 in results clicked by the user U_1
$SourceTrustScore(U_1, S_1)$	Getting the trust score for data source S_1 from the user U_1 perspective
$ResultLikeScore(U_1, P_1, Q_1)$	Getting the like score of a result predicate P_1 for the user U_1 when entered the query Q_1

Below, we explain each function with more details.

$QueryTermFrequency(U_1, QT_1)$ measures the frequency of certain words in the user's queries where: $QueryTermFrequency(U_1, QT_1) = Function(queryTerms(U_1, Q_1, QT))$.

The function returns the count of a specific query term repeated in the user's queries. The higher frequency of a certain word in the user's queries indicates that this word is more important to the user and hence can be used to enrich the queries. Figure 19 shows a representation of the user's term frequencies.

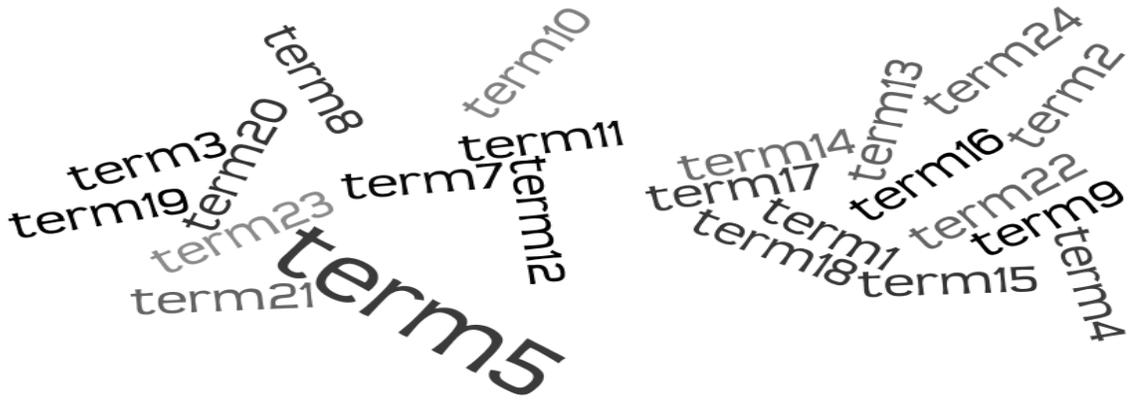


Figure 19 Example of Terms Frequency

$RDFTermLikeScore(U_1, RT_1)$ measures if the user likes a certain RDF term when the user browses the results. Its function combines different measures taken while the user interacts with the results to create a score for the RDF term. Below we show how we calculate the score of a particular RDF term.

$$RDFTermLikeScore(U_1, RT_1) = Function(like(U_1, Q_m, RT_1), dislike(U_1, Q_m, RT_1)) = (1 - dislike(U_1, Q_m, RT_1)) * like(U_1, Q_m, RT_1)$$

where the RDF term liked and disliked in query Q_m . and the maximum value of the function is limited to 1.

$SourceTrustScore(U_1, S_1)$ considers if the user rated the source previously as a trusted source and gives it more weight if so. Then it sorts the results based on the maximum score.

$$SourceTrustScore(U_1, S_1) = function(trustMore(U_1, Q_1, S_1)) = \sum trustMore(U_1, Q_1, S_1)$$

$ResultLikeScore(U_1, R_1, Q_1)$ combines different interactions by the user such as explicit and implicit feedback such as printing, bookmarking, clicking, and visit durations. Below is the equation for this function:

$$\begin{aligned}
 \text{ResultLikeScore}(U_1, R_1, Q_1) = & \text{Function}(\text{filter}(U_1, Q_1, R_1), \text{showFirst}(U_1, Q_1, R_1), \\
 & \text{resultClicked}(U_1, Q_1, R_1), \text{resultVisitDuration}(U_1, Q_1, R_x, VD_1), \text{resultPrinted}(U_1, Q_1, R_1), \\
 & \text{resultBookmarked}(U_1, Q_1, R_1)) = (1 - \text{filter}(U_1, Q_1, R_1)) * (10 * \text{showFirst}(U_1, Q_1, R_1) + \\
 & 0.25 * \text{resultClicked}(U_1, Q_1, R_1) + 0.25 * \text{resultPrinted}(U_1, Q_1, R_1) + 0.25 * \\
 & \text{resultBookmarked}(U_1, Q_1, R_1) + 0.125 * \text{resultVisitDuration}(U_1, Q_1, R_x, VD_1))
 \end{aligned}$$

4.5 Semantic Query Manipulation Agent

The semantic query manipulation agent is required to manipulate semantically, enrich, and process the user's queries. It interacts with different agents to process the user's queries. It interacts with the interface agent to acquire the user's query to manipulate it through different steps. Moreover, the user's query might miss some information that needs to interact with the interface agent to revise the query. After the query is manipulated, the semantic annotated query is sent to the personalized retrieval agent. The semantic query manipulation functions are shown in Figure 20.

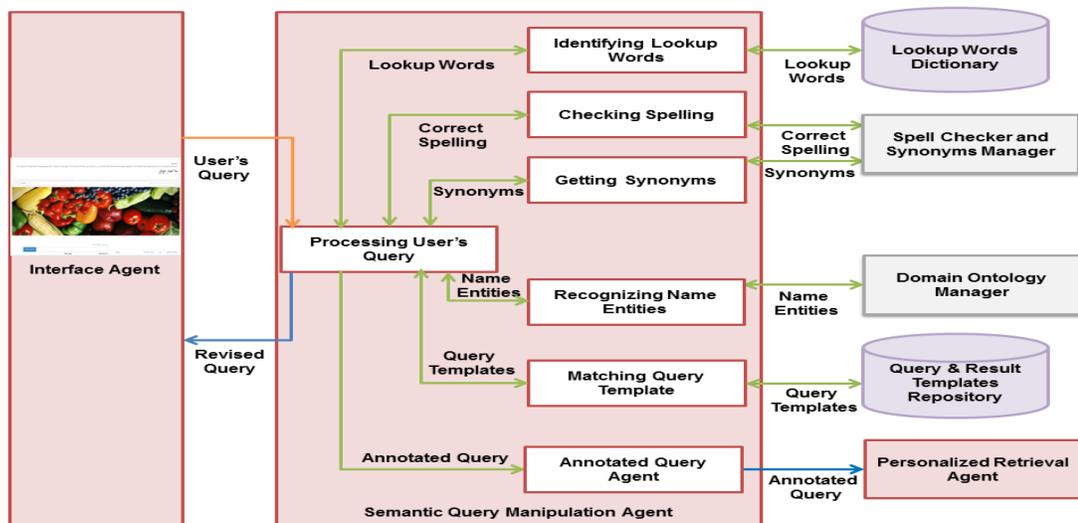


Figure 20 Semantic Query Manipulation Agent Functions

We present the required steps of the semantic query manipulation. After getting the user's query, we first identify the query's language because each language has its own syntax, linguistic characteristics, and way of processing. Then we tokenize the query into tokens (terms) and check the correct spelling of the query using the spell checker and synonyms manager services, which check the term spelling and provide synonyms for any term.

After that, we do part-of-speech (POS) analysis to identify which term is a verb, which term is a noun, and so on. Then the query is classified into the appropriate question type using the lookup words dictionary services. Lookup words dictionary is a repository for the predefined terms that help in recognizing the stop-words, the question types, the relations between terms, and other predefined names. It is a multilingual dictionary that contains a list of terms for each language to be used in looking up and finding the matching terms. The question type is required to decide how the answer will be formulated.

Next, we identify the measurement keywords within the terms of the query. We look for (1) numbers that represent quantities, (2) measurement units, and (3) other measurement means such as serving sizes in the food and health domain. An example of the question is, "What food can provide me 100 mg of calcium?" For measurement units and measurement means, we use the lookup words dictionary, which contains predefined keywords to compare terms with.

After that, noise words are filtered using lookup words dictionary's list of noise words, such as *do*, *does*, *an*, *the*, etc. This helps in limiting the processing to only the words that could be related to the domain ontology.

Then we identify the terms related to the domain ontology through the named entity recognition techniques. The agent interacts with the domain ontology manager to get a

populated list of all ontologies' concepts and knowledgebases' instances. It determines whether the query term is a concept or an instance and gets the semantic information for the identified term.

Next, we use a morphological analysis tool to get the root of the remaining words, which will help us in correlating the remaining terms. After that, we identify the possible relations between these terms using the lookup words dictionary to match it with the pre-defined relationship terms. We then identify other terms that are defined in the repository, such as the terms that mean the user is asking for their daily needs. An example is, "What food can provide me *enough* calcium?"

After that, we check the terms that we could not match. For this, we use techniques such as context analysis, pattern, and synonymous to identify the nearest match. The agent also gets the synonyms of the terms from the spell checker and synonyms manager to match them with the terms of the user's query.

Ambiguity could happen in the previous steps, such as when the term "cholesterol" is classified as both nutrition and disease. Another example is when we identify two relations when we have only two concepts or instances and only one of these relations is correct. For this, we analyze the context of the query, use the pattern for the previously known cases, and get weighted named entity recognition to judge on the most likely correct match.

Then we match the identified terms with the best query template using the query and result templates repository services. The query template models the possible input from the user with the respective semantic queries and the expected results. The query and result templates repository stores the query templates. If there is no template that can be matched with the user's query, then the agent revises the query using the user's profile

first. If the user’s query is still not matching any query template, then the agent interacts with the user to receive for more information.

The user’s profile is retrieved from the user’s profile agent for two reasons: to revise the query and to enrich the query with more information about the user. Finally, a semantic annotation of the query is produced and sent to the personalized retrieval agent. Figure 21 illustrates these steps graphically.

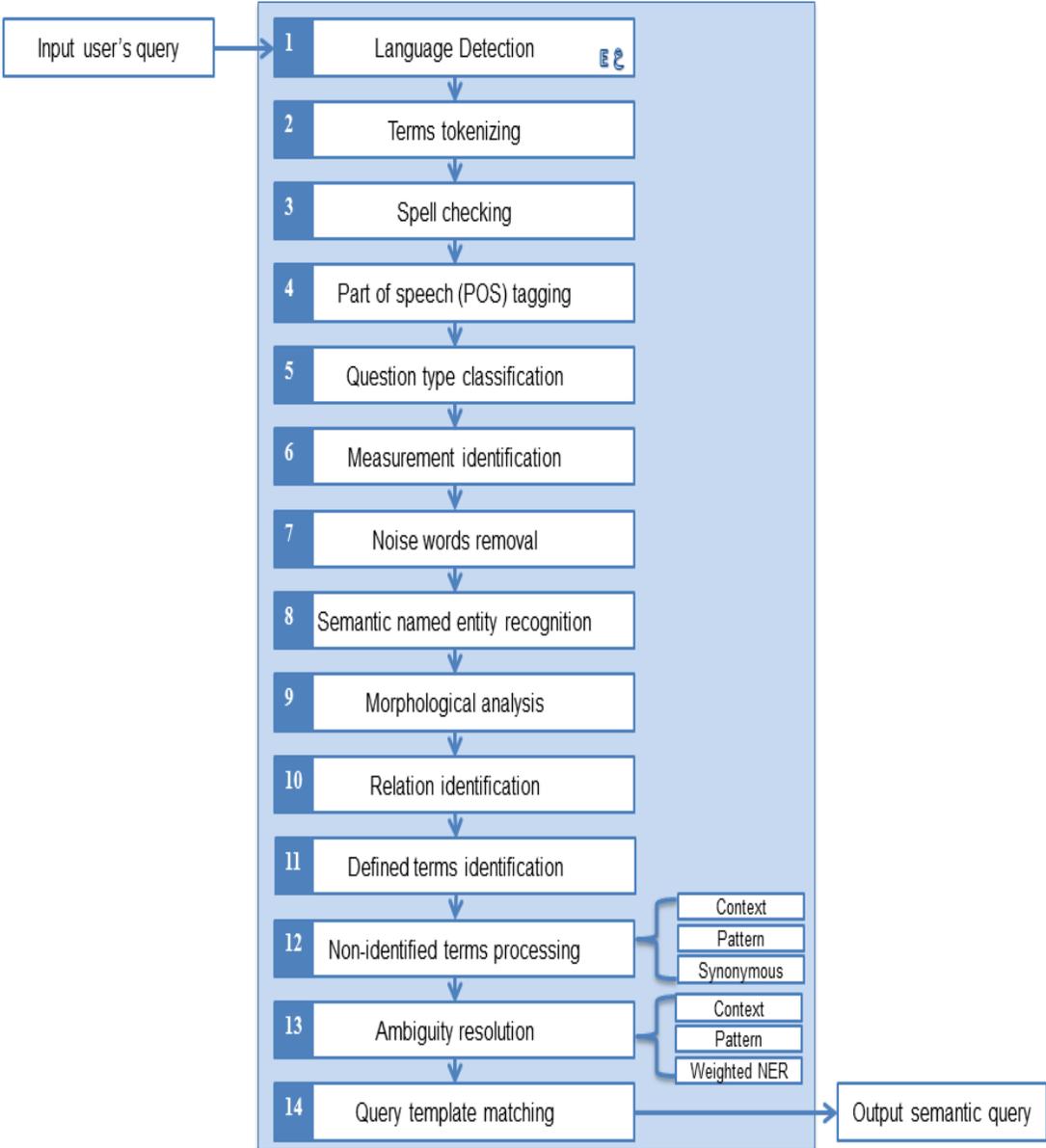


Figure 21 Semantic Query Manipulation Steps

Since we are not doing natural language processing (NLP), we must define specific query templates to scope the user's queries and match them to the related ontologies. Query templates, in our research, represent all expected queries from the user, define the concepts that could be extracted from the user's query, correlate different ontologies that are required to answer the query, and finally specify the answer template for each query. Matching the user's query to the predefined query templates is not binary matching; it is more complicated. Identifying the concepts and relations within the user's query that are related to the domain ontology is not sufficient to match them with any query template. We try to fill the most appropriate query template concepts and relations, which were identified in the query-processing phase. However, there are some cases where we have incomplete information and hence we need to depend on other sources to fill the query template. After getting all what we can extract from the query, we get aid from the domain ontology to detect the missing information based on what is found. Then we look at the user profile, if any, and fill in the missing information from the profile properties. Finally, we can go back to the user and ask explicitly for more information to match the query template.

Figure 22 outlines the algorithm used for query template matching. It takes threshold as input to return the best matched query above that threshold. So, even if there is no query template matching 100%, it will return the best matched based on the threshold.

ALGORITHM	matchTemplate	
Input	QSTP	Question Type
	SN	Array of identified Semantic Name Entities
	OT	Array of identified other terms
	thrSM	Threshold for matching Semantic Name Entities
	thrOT	Threshold for matching other terms
Output	QRTM	Matched query template (if any)
1	Initiate:	
	QRTMS	Array of defined query templates
	QRTM	Query template (set empty by default)
	CMOT	Similarity score for Other Terms
	CMSN	Similarity score for Semantic Name Entities
	CMOTS	Array for Similarity scores for Other Terms on all Query Templates
	CMSNS	Array for Similarity scores for Name Entities on all Query Templates
	CRSP ₁	Arrays of the correct spellings of a term (temporary holder)
	OT	Array of identified other terms
	QRTM	Query template
	NeedRevise	Boolean of default False (determine if the query needs revising with the user)
2	Set x = 1	
3	Loop	For each QRTM _x ∈ QRTM where 1 ≤ x ≤ n (for each query template QRTM _x in n templates)
4.1	CMOT = compare(OT, QRTM _x)	- returns the similarity for OT
4.2	CMSN = compare(SN, QRTM _x)	- returns the similarity for SN
4.3	Push CMOT to CMOTS array	
4.4	Push CMSN to CMSNS array	
5	End Loop	
6	Set iOTMax = index (getMaxScore(CMOTS))	
7	Set sOTMax = score(CMOTS(iOTMax))	
8	Set iSMMMax = index (getMaxScore(CMSMS))	
9	Set sSMMMax = score(CMSMS(iSMMMax))	
10	If (sSMMMax ≥ thrSM and sOTMax ≥ thrOT and iOTMax = iSMMMax)	
11	QRTM = QRTMS(iSMMMax)	
12	Else	
13	Sort(CMSNS)	
14	Sort(CMOTS)	
15	Set y = 1	
16	Loop	for each CMSN _y ∈ CMSNS where 1 ≤ y ≤ o (where there are o objects in CMSNS)
16.1	Loop	for each CMOT _z ∈ CMOTS where 1 ≤ z ≤ p (where there are p objects in CMOTS)
16.2	If(score(CMSN _y) ≥ thrSM and score(CMOT _z) ≥ thrOT and y = z)	
16.3	QRTM = QRTMS(y)	
16.5	End If	
16.6	End Loop	
16.7	End Loop	
17	End If	
18	Return QRTM	
19	End	

Figure 22 Outline for Template Matching Algorithm (matchTemplate)

4.6 Personalized Retrieval Agent

The Personalized Retrieval Agent is required to personalize the retrieved results. It gets the annotated query from the semantic query manipulation agent and then identifies the results template that defines the expected results of the query. It then retrieves and personalizes the semantic result before sending it to the interface agent, where it is formulated for the end user. The personalized retrieval agent functions are shown in Figure 23.

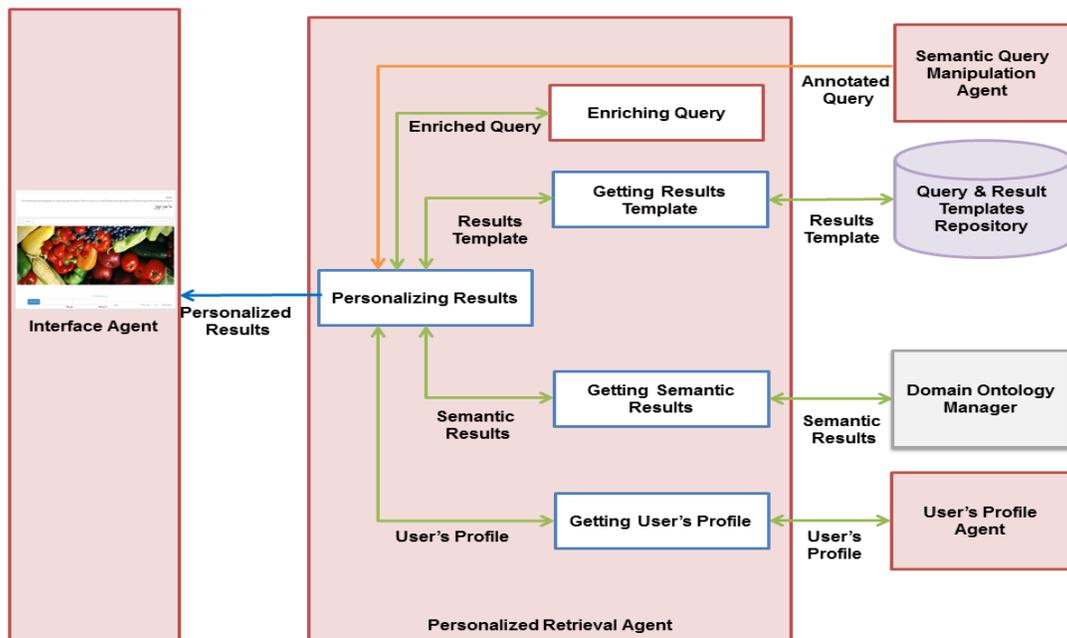


Figure 23 Personalized Retrieval Agent Functions

The personalized retrieval agent communicates with different agents to process the user's annotated query and personalizes the retrieved results. First, it receives the annotated query from the semantic query manipulation agent. It then enriches the query based on the user's profile. After that, it determines the appropriate results template that matches the annotated query. The results template determines the semantic queries needed to be

reasoned to get all the expected results. After that, the semantic results are post-processed to determine if there are any conflicts or possibilities for aggregating similar results, and then ranks and sorts them after getting the user's preferences from the user's profile agent. Finally, the results are personalized and sent to the portal agent to show them to the user. The steps are illustrated in Figure 24.

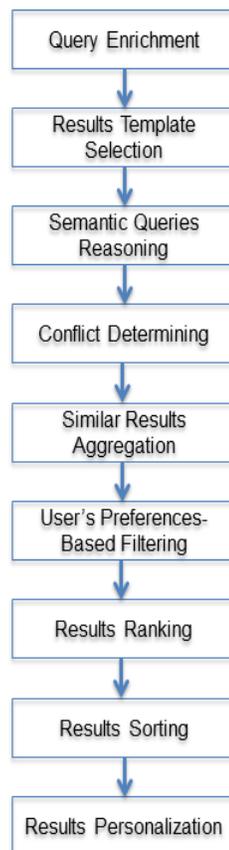


Figure 24 Results Personalization Steps

We go through the results personalization steps in more detail. First, the selection of the results template is correlated to the query template. Each query template corresponds to a results template. The query template defines the possible input from the query, and the results template defines the expected output results. The result template contains a list of

semantic queries that need to be reasoned by the domain ontology manager, which provides a number of services, such as reasoning and executing the semantic queries. The domain ontology manager sends the output as semantic results in RDF format.

Next, the conflicts between results are determined. In the Semantic Web, any relation in the statement in the web source is represented as a triple <subject, predicate, object>. Each record in the results has RDF terms and relations between each two terms. This is represented in RDF by the triple <subject, predicate, object> denoted <S, P, O>. We define a conflict if two results have an inverse relationship between the same terms. After that, the agent aggregates similar results. Then we get the user's profile and filter the results based on the user's interactions with previous results. For example, if the user trusts a particular source more, then we show the results from that source first. We rank the results based on the user's preferences and then sort them based on the results' aggregation and user's preferences as modeled in the below function.

$$\textit{Sort}(\textit{results}) = f(\textit{aggregated results}, \textit{user's preferences})$$

We give more weight to the aggregated results where the same relation is repeated in more web sources. Also, we give weight to the user's preferences to sort them first. Then the personalized results are sent to the interface agent to display them. More details are offered in Chapter 6.

CHAPTER 5

CAPTURING USER'S PREFERENCES RELATED TO FOOD AND HEALTH

Our research objectives are to manipulate semantically the user's query and then personalize the results in the domains of nutrition, food, and health. The user's profile represents the user's preferences. In the following sections, we first analyze and capture the user's preferences and then propose a user's profile that represents these preferences. Therefore, one milestone in constructing the user's profile is to capture the user's food preferences and health conditions. This will help in answering the user's queries with more relevant results based on the user's personal preferences, health condition, culture, religion, etc.

5.1 Methodology

To capture the user's preferences, we start with some motivation questions that help in driving the attributes of the user's preferences. Second, we analyze and study the answers of the motivation questions to identify the attributes that affect the user's choices and preferences. For this, we first classify the identified attributes into categories of attributes, then we study the relationship between these attributes and see whether it is possible to combine them or resolve any conflict between them. We propose to give priority and

weight for each attribute to capture the influence of these attributes on the user. Figure 25 illustrates the methodology to capture the user's preferences.

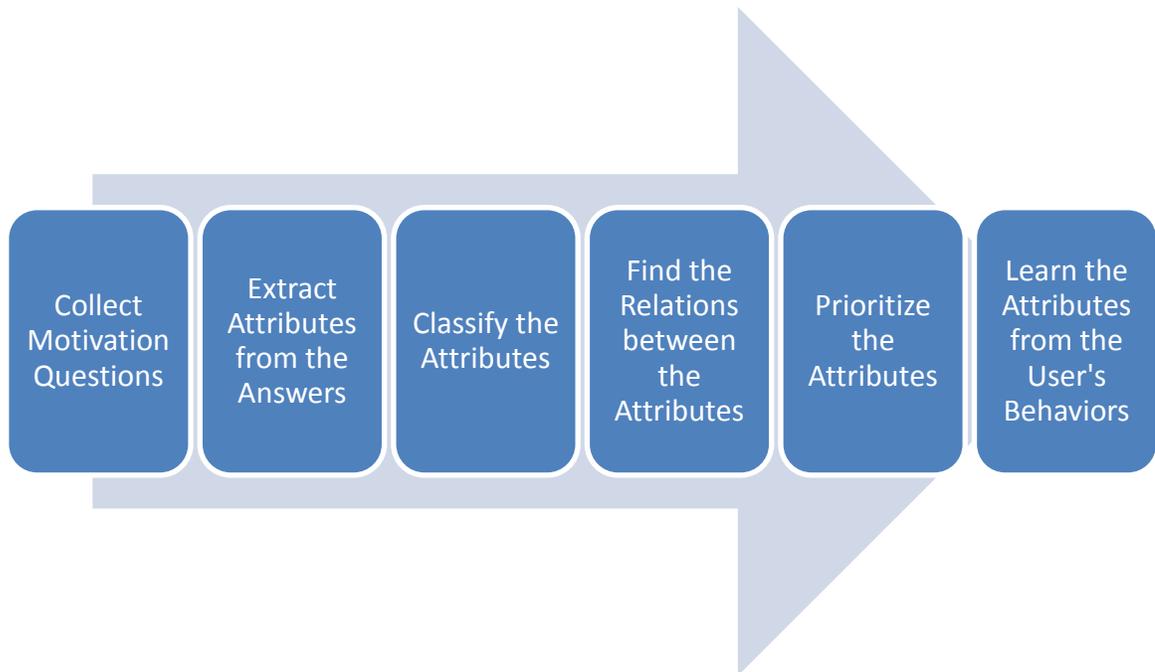


Figure 25 Methodology to Capture User's Preferences

5.2 Motivation Questions

We start this section by raising questions that help in identifying the attributes affecting the user's food choices and preferences.

- Why do we like a specific food to eat?
- Why do we avoid a certain food?
- What attributes could we extract from the answers for the first two questions?
- Is there any relation between these attributes? (e.g., Is there any relation between the locations or culture with regard to the food preferences?)
- Can these attributes be combined?

- Do these attributes conflict with each other?
- Is there an order or weight for these attributes?
- Are we committed to our culture and religion when it comes to food preferences?
(e.g., Is it ok to try new food in a new culture?)
- Are we committed to health constraints when it comes to food?
- Are we considering the daily nutrition needs when we choose the food?
- How can we know and learn others' food preferences and constraints? Are they written somewhere?

5.3 Attributes Affecting the Choice of Foods

To study how we prefer a specific food and do not prefer another, we also need to study the attributes that influence our food choices. To come up with these attributes, we analyze the answers to the previous questions. Then we classify these attributes into four categories: personal preferences, health conditions, cultural preferences, and religion restrictions. Below is a detailed explanation for each category followed by other attributes that could be considered in future work.

5.3.1 Personal Preferences

Many people prefer certain foods while disliking other foods with and without reasons in mind. There are many examples of the possible reasons for preferring or avoiding some food such as the taste, look, color, and smell of the food. And then sometimes we do not know why we like or dislike a certain food, as it might be a personal habit, such as when we do not eat a certain food while children and then we get used to avoiding it.

5.3.2 Health Condition Constraints

Health conditions can restrict some food or limit their quantity while encouraging other food. Examples include relations between some diseases and certain foods. Some foods can help in preventing or treating diseases, such as eating oranges to help treat the ‘flu. There are also food allergies that can cause serious impact on people’s health. Another example is when a woman is pregnant and she is advised to eat healthy foods that contain vitamins, minerals, and so on.

Other health attributes can be the health goals, daily needs, and commitment to a sport or diet program limiting food choices. A health goal can drive someone’s daily food program, such as in reducing weight. There are many food programs to lose weight, and they specify different food types and quantity on daily or meal-by-meal basis. Having a health goal can restrict many types of food and motivate taking other types.

5.3.3 Cultural Preferences

Different cultures come with different customs and traditions. Culture, location, and language are correlated when we talk about culture. Our focus in this thesis is on including specific aspects of culture related to food selection, such as (1) what food is acceptable in a certain culture and what food is not acceptable; (2) what food is preferred in a certain culture and what food is not preferred; (3) what popular nutrition is used by a certain culture; and (4) what recipes are commonly used by a certain culture. As an example from Saudi Arabian culture, Saudis prefer eating rice that comes in different colors and flavors, such as red, white, or brown, and normally it is cooked with meat, fish, or chicken.

The relationship between food and culture also shows when we recommend foods across different cultures, which may involve different measurements and different recipes. Certain foods are substituted with other foods in different cultures, as are food combinations, the timing of meals, eating certain foods at certain events, and finally the different names cultures give to the same foods.

People's culture can be correlated to their location, both original and new. A person's location of origin is a factor, as is how rigidly that person follows their food culture. This can be determined by monitoring the person's interaction and behaviors with food recommendations. That same person's current location and how open he or she is to the different foods in the new location could also be inferred from reactions with the food recommendations. For example, a Muslim Saudi woman greatly influenced by her place of origin visits Japan and does not eat sushi or other Japanese food. For another example, a different Muslim Saudi is not highly influenced by his place of origin, and when visiting Japan he tries many Japanese foods.

Another way is to look at the relation between food and culture is to consider the time dimension. Some foods are preferred at breakfast, while others are preferred at lunch and dinner. Many restaurants, for example, offer appropriate foods based on the time of the day. This will restrict the food choices available. Also, there are some special days, an issue of both culture and date, when the offered food is different. A good example is the month of Ramadan, which for most Muslims has its unique foods. Therefore, we can take an example of Muslims' culture of food during the Hijri month of Ramadan, when Muslims fast throughout the entire month. However, we cannot say this about the Muslim culture in general as it is different from location to location. This shows how time and location have different effects on culture. Not all foods eaten in Ramadan are used in

other months. Let us say the preference for such foods during Ramadan is high, while it is lower in other months.

5.3.4 Religious Constraints

Some religions have food restrictions so it is important to capture these constraints to avoid inappropriate food advices. As an example, in the Islamic religion alcohol and pork are prohibited, so it is not acceptable to recommend any food that contains alcohol or pork to Muslims. There are other examples from other religions as well.

5.3.5 Other Attributes

We know of other attributes that could be considered, but we do not have data to support them at this time. We defer them to future work. Below is the summary of two attributes with examples.

The first attribute is the current climate condition where different foods are good for different seasons. Climate affects food choices, such as summer fruits and winter fruits and preferring cold food and drink, such as ice cream and cold water, in the summer while preferring a cup of coffee or tea in the winter.

A second attribute is a person's financial state, where a budget may restrict expensive foods.

5.4 Relations between the Attributes

Choosing the right food thus depends on many attributes that sometimes conflict with each other. For example, someone who has diabetes may like sweet food. Would this person give the health condition or personal preferences the priority? Maybe in this example it is wise to give the health condition the priority. However, there are cases that

are more complex where conflicts that are more difficult could happen between different attributes. We try to analyze the relations between these attributes and highlight the possibilities of conflicts and combinations between them.

5.4.1 Combinations

Some attributes could be combined without conflicts. A good example is when healthy people eat what they like while maintaining a quantity limit and a balance between different foods for good nutrition while exercising. Only the person knows how to combine these attributes in the best way, and that is why we give the user the flexibility to define priorities in these attributes.

5.4.2 Conflicts

Conflicts between the attributes can abound and the possibilities are high. A good example is when a person committed to a diet program is invited to a wedding reception serving heavy food. This is well-known conflict in some cultures such as Saudi Arabia. The challenge is how to deal with this conflict, and we think it is difficult to automate a process to resolve such conflicts. That is why we need to analyze these attributes and prioritize them based on each person's preferences and priorities.

5.4.3 Order and Weight

Based on our analysis in the previous sections, the user should prioritize the attributes that affect the choice of foods. People are responsible for their choices, especially in the lifestyle and food they want to eat. The proposed approach is to give users different attributes that affect their food choices so they can prioritize them based on their own judgment. Then we calculate a weight for each attribute by combining their preferences

and priorities to come up with a recommendation equation that helps us to recommend the best food for them, as follows:

Given attributes $A_1, A_2, A_3, \dots, A_m$

Given priorities $P_1, P_2, P_3, \dots, P_n$

Given weights $W_1, W_2, W_3, \dots, W_k$

Users are given the attributes $(A_1, A_2, A_3, \dots, A_m)$, and then they select the priorities for each one $(P_1, P_2, P_3, \dots, P_n)$. After that, the system calculates the weight for each attribute as follows: $W_1 = 1 / P_1, W_2 = 1 / P_2, W_3 = 1 / P_3, \dots, W_k = 1 / P_n$

The weights are used once for ranking the results in a combination with other factors that will be discussed in personalizing the results.

5.5 Survey on the Attributes and Their Priorities

We conducted a survey to determine the attributes that affect the user's food choices and their priorities to establish a base line and default attributes with their order and weights. We first created a survey form with all the questions in the survey and published it online. A month passed while people filled out the form, following a link that collected their responses. We studied the responses to determine possible attributes and their rank. This helped us in determining the default weight for each attribute.

The developed survey was sent to a collection of people from which we have collected responses from 142 professionals living in Saudi Arabia ranging in age from 25 to 45 years. They stated they used the Internet and popular search engines daily to find information. They spoke English as second language. They responded with ranking the

attributes in the survey and suggesting additional attributes. The responses are summarized in TABLE 7.

TABLE 7 Responses for Survey on the Attributes and Their Priorities

		Scale and Responses				
No	Attributes affect the choice of a food	Very Important	Important	Normal	Not Important	Not Applicable
1	Personal preferences	67	26	15	16	18
2	Health condition and/or restrictions	76	31	8	0	27
3	Cultural-based preferences and/or restrictions	38	29	18	21	36
4	Religion-based preferences and/or restrictions	58	34	23	3	24

Based on the results, we can say the health condition is the most important attribute that affects the food choices for those who answered the survey, followed by personal preferences, religion restrictions, and finally cultural preferences.

5.6 Learning the Attributes from the User's Behaviors

We have identified the possible attributes that could affect food choices in the previous sections. We then need to learn and capture these attributes for food preferences from the user. We have different ways to know the user preferences. One way is to give users a

form to fill out and let them express what they prefer and prioritize these attributes. Another way is to let the system learn users' preferences based on their interactions with the system. For example, the system might observe that the user always selects recipes that contain tomatoes, and then the system could conclude that the user likes tomatoes. User will be asked to review and confirm such conclusions about their food preferences. We also give users the chance to prioritize these preferences so that ASPIR can give them better recommendations.

CHAPTER 6

MODELING OF THE FRAMEWORK PROCESSES

This chapter presents the details of models used to represent the framework processes. This includes the models used for the annotated knowledgebase, semantic predicates, query, retrieved results, user's interactions with the results, user's profile, query enrichment, results conflict resolution, and personalization.

6.1 Modeling of the Knowledgebase and Predicates

We use the notation H to represent the set of A health conditions and the notation F to represent the set of B foods. We use the notation D to represent the set of N data sources that are annotated in the knowledgebase, where each data source contains predicates with relations between the health conditions and foods. Next, we show the model used for the annotated knowledgebase.

$$H = \{h_a \mid h_a \text{ is a health condition, } a = 1, 2, \dots, A\}$$

$$F = \{f_b \mid f_b \text{ is a food, } b = 1, 2, \dots, B\}$$

$$D = \{d_n \mid d_n \text{ is a data source, } n = 1, 2, \dots, N\}$$

We use the notation S to represent the set of I subjects and the notation O to represent the set of J objects, while the notation P represents the set of K predicates. Each data source d_n contains a set P_{d_n} of annotated predicates in the knowledgebase. The predicates are the relations between the subject and the object. For example, if, "An apple is good for diabetes," is a statement in the data source, then the subject is "apple," the

object is “diabetes,” and the predicate is “is good for.” The predicates can be categorized into two groups: positive predicates denoted as P^+ and negative predicates denoted as P^- . In the domains of food and health, examples of the positive predicates are: “has positive effect on,” “treats,” and “prevents.” Examples of negative predicates are: “has negative effect on” and “causes.” We use the notation R to represent all relations between the subject set S and the object set O where every ordered pair (s_i, o_i) in R corresponds to a set of predicates P_i that is a subset of P . Next, we show the model used for predicates.

$$S = \{s_i | s_i \text{ is a subject, } i=1,2,\dots,I\}$$

$$O = \{o_j | o_j \text{ is an object, } j=1,2,\dots,J\}$$

$$P_{d_n} = \{p_{d_{nk}} | p_{d_{nk}} \text{ is a predicate from the data source } d_n, k=1,2,\dots,K\}$$

$$P_{ij} = \{p_{d_{nk}} | s_i \text{ is related to } o_j \text{ by the predicate } p_{d_{nk}} \text{ in } d_n, k = 1,2, \dots, K, n = 1,2, \dots, N \}$$

$$P = \cup P_{ij}$$

$$P_{ij}^{++} = \{p_{d_{nk}} | p_{d_{nk}} \in P_{d_n}, p_{d_{nk}} \text{ is a positive predicate}\}$$

$$P_{ij}^{--} = \{p_{d_{nk}} | p_{d_{nk}} \in P_{d_n}, p_{d_{nk}} \text{ is a negative predicate}\}$$

$$R = \{(s_i, o_i) | s_i \text{ is related to } o_i \text{ by } P_{d_{nk}} \text{ in } d_n, k=1,2,\dots,K, n=1,2,\dots,N\}$$

6.2 User’s Profile Model

We use the notation U to represent the set of M users as follows:

$$U = \{u_m | u_m \text{ is a user, } m=1,2,\dots,M\}.$$

We use the notation UP_m to represent the user's profile for the user u_m , which consists of five parts: (1) preference elements, notated as PE_m , (2) the cultural preference of the user, notated as CP_m , (3) the values of the profile's basic information, notated as VBI_m , (4) the data source satisfaction evaluated by the user, notated as DS_m , and (5) the preferred predicates based on the user's interactions, notated as PP_m . We explain and model each one of these five parts below.

First, each user's profile contains a set of preferred elements where each user likes and dislikes certain foods and has a specific health condition. We use the notation PE_m to represent the set of X preferred elements that belong to either the health condition domain H or food domain F . We use the notation VPE_m to represent a function that sets the value of each preferred element. It ranges from -1, which means it is not preferred, 0, the default value that has no opinion, and +1, which means it is preferred. We use the notation WPE_m to represent a function that determines the weight of each preferred element. It ranges from 0 to 1, which reflects the importance of each preferred element. We use the notation $wvpe_{mx}$ to represent a function that determines the total weight for each preferred element by multiplying the weight and the value of the preferred element. Below is the model for the preferred elements.

$$PE_m = \{pe_{mx}, x = 1, 2, \dots, X, pe_{mx} \text{ is preference element for the user } u_m\}$$

$$PE_m \subseteq H \cup F$$

$$VPE_m: \text{function for value of the preference element } pe_m$$

$$VPE_m: PE_m \rightarrow \{-1, 0, 1\}$$

$$WPE_m: \text{function for weight of preference element } pe_m$$

$$WPE_m: PE_m \rightarrow [0, 1]$$

$$WPE_m(pe_{mx}) = wpe_{mx}$$

$wvpe_{mx}$: function to calculate the total weight for each preference element pe_x for

user u_m

$$wvpe_{mx} = wpe_{mx} \cdot vpe_{mx}$$

Second, each user's profile contains a set of culture preference where each user belongs to a certain culture corresponding to some food preferences. We use the notation CP to represent a set of C culture preferences. The culture preference in our context is related to the preference of food F . For example, CP_{cb} represent the cultural c preference, such as Saudi, for the food b , such as rice. Each user u_m can have a cultural preference notated by CP_m , which is part of CP . We use the notation VCP_m to represent a function that determines the value of the cultural preference. It ranges from 0 to 1. If the user belongs to a specific culture, then the value is 1. If the user does not belong to that culture, then the value is 0. We use the notation WCP_m to represent a function that determines the weight of the cultural preference. It ranges from 0 to 1, and the value of the weight reflects how important the cultural preferences are to the user and how they affect the user's choice of foods. For example, if a Saudi person visits Japan and does not eat sushi and other Japanese foods and likes to have only Saudi food, then the Saudi culture has great influence on the user's food choices. However, if the user has less influence from the Saudi culture, then the user will be open to trying many Japanese foods. Below is the model for the user's preferred culture.

$CP = \{CP_{cb} \mid CP_{cb} \text{ is the cultural preference value of culture } c \text{ on food } b \text{ such that } c = 1, 2, \dots, C \text{ cultures, } b = 1, 2, \dots, B \text{ foods}\}$

$CP_m \subseteq CP$, is the culture preference for the user U_m

VCP_m : function for value of the culture preference CP_m

$VCP_m: CP_m \rightarrow \{-1, 0, 1\}$

WCP_m : function for the weight of culture preferences for user U_m

$WCP_m: CP_m \rightarrow [0, 1]$

Third, each user's profile contains a set of basic information about the user, such as weight, length, and blood type. We use the notation BI to represent a set of E variables for the basic information. We use the notation VBI_m to represent a function that determines the constant values of the basic information BI_m , which are set by the user u_m . Below is the model for the user's basic information.

$BI = \{bi_e | bi_e \text{ is a variable for basic information, } e=1,2,\dots,E\}$

$bi_1 = \text{height}, bi_2 = \text{weight}, \dots \text{etc}$

$VBI_m = \{VBI_{mbi} | VBI_{mbi} \text{ is the constant value of the basic information variable } bi_i \text{ set by the user } u_m\}$

Fourth, we use the notation DS to represent a function for the user's explicit satisfaction on a specific data source. Each user has different viewpoint about the data sources, and this leads to different levels of satisfaction. The values of the function DS range from -1, which means the user is not satisfied with the data source, 0, which is the default neutral value, and +1, which indicates high satisfaction about the data source. Below is the model for the data source satisfaction.

DS_m is the data-source satisfaction function for the user u_m

$DS_m = \{DS(d_k, u_m) | d_k \in D, u_m \in U, d_k \text{ is a data source that has been evaluated by } u_m\}$

$DS: (d_k, u_m) \rightarrow \{-1, 0, 1\}$

$$DS(d_k, u_m) = ds_{km}$$

$$ds_{km} = \begin{cases} +1, & \text{if } d_k \text{ is more satisfied for } u_m \\ 0, & \text{if } d_k \text{ is normal satisfied for } u_m \\ -1, & \text{if } d_k \text{ is not satisfied for } u_m \end{cases}$$

Fifth, we use the notation PP_m to represent the set of preferred predicates for the user u_m , which is the union of all weighted predicates based on the user's interactions. More elaboration on the preferred predicates will be discussed when we talk about the user's interaction with the results in Section 6.3.

Finally, the user's profile UP_m is the union of the five parts: preference elements PE_m , the cultural preference of the user CP_m , the value of the profile's basic information VBI_m , the data source satisfaction evaluated by the user DS_m , and the preferred predicates based on the user's interactions PP_m . Below is the model for the user's profile.

UP_m is the user profile set for the user u_m

$$UP_m = PE_m \cup CP_m \cup VBI_m \cup DS_m \cup PP_m$$

6.3 User's Interactions Modeling

Users interact with the results, and we capture both explicit and implicit interactions. We have eight measures, four explicit and four implicit. The first explicit measure is the data source satisfaction DS_m , which was explained in Section 6.2. The remaining seven measures are related to the resulting predicates. The explicit ones are rating the result, marking a certain result to show first, and marking a specific result to be filtered out. The implicit measures are the time of visiting a certain result, clicks on a certain result, and printing and bookmarking a specific result. We use the notation g to represent these seven measures. We use two functions to represent the user's interactions with the results. We

use the notation WPP_g to represent a function that determines the weight of each measure g . It ranges from 0 to 1 based on the measure's rank and user's priority. For example, if printing the result is not important to the user, then the value will be 0. We use the notation VPP_g to represent a function that determines the value of each measure g . It ranges from 0 to 1, where 0 means the user does not show interest and 1 means the user shows the interest in the result. We use the notation wp to represent a function that calculates the weight for each predicate by adding all seven measures in addition to the eighth measure, the data source satisfaction, and then dividing their sum by 8. Based on the predicate type, we give either a positive sign for the positive predicates or a negative sign for the negative predicates. We use the notation PP_m to represent the set of preferred predicates for the user u_m , which is the union of all weighted predicates based on the user's interactions. Below is the model for the user's interactions with the results.

$$VPP_g(u_m, p_{d_{nk}}) = vpp_{m g p_{d_{nk}}} \in [0, 1], g=1, 2, \dots, 7$$

$$WPP_g(u_m) = wpp_{mg} \in [0, 1], g=1, 2, \dots, 7$$

wp : function which determines the weight of a predicate

$$wp_{m p_{d_{nk}}} = \begin{cases} \frac{1}{8}(ds_{km} + \sum_{q=1}^7 (wpp_{mq} \cdot vpp_{m g p_{d_{nk}}}) , & \text{if } p_{d_{nk}} \in P_{ij}^+ \\ \frac{-1}{8}(ds_{km} + \sum_{q=1}^7 (wpp_{mq} \cdot vpp_{m g p_{d_{nk}}}) , & \text{if } p_{d_{nk}} \in P_{ij}^- \end{cases}$$

$$PP_m = \cup wp_{m d_{nk}}$$

6.4 User's Query Model

A user can have many queries. We use the notation Q_m to represent set of L queries performed by the user u_m . The notation Q_{ml} represents a query l that is performed by the

user u_m and contains a set of W words where we use the notation q_{mlw} to represent a word w in the query Q_{ml} . The words that concern us should belong to either the annotated knowledgebase of the food and health domains or the possible relations between them notated by predicates. For any word in the query that is not matched with the knowledgebase and relations, we use synonyms to match. If no synonym matches, then we ask the user either to clarify the word or use an alternative one. Finally, if we are not able to match it, then we show that the query has no results. Below is the model for the query.

$$Q_m = \{Q_{ml} \mid Q_{ml} \text{ is query for user } u_m, l=1,2,\dots,L \text{ number of queries for user } u_m\}$$

$$Q_{ml} = \{q_{mlw} \mid q_{mlw} \text{ is a word in the query } Q_{ml}, q_{mlw} \in H \cup F \cup P, w = 1,2,\dots,W \text{ number of words in query } q_{ml} \text{ for user } u_m\}$$

6.5 Query Enrichment Model

In query enrichment, we find the set of words that we can add to enrich the query based on the user's profile. We use the notation Q_{ml}^+ to represent the additional words used to enrich the query Q_{ml} , which was performed by the user u_m . These additional words are retrieved from the user's profile, specifically the set of X preference elements of the user pe_{mx} . We use only the top preference elements that have weight, notated by $wvpe_{mx}$, more than a specific threshold β . We use the notation Q_{mi}^{enr} to represent the enriched query, which is the union of the user's query Q_{ml} and the additional words Q_{ml}^+ that have high weight in the user's profile. Below is the model for the query enrichment.

$$Q_{ml}^+ = \{pe_{mx} \mid pe_{mx} \in PE_{mx}, wvpe_{mx} \geq \beta, \beta \text{ is a constant (threshold)}, x = 1,2,\dots,X\}$$

$$Q_{mi}^{enr} = Q_{mi} \cup Q_{mi}^+$$

6.6 Retrieved Results Modeling

After the semantic manipulation of the user's query, the query is reasoned and the results are retrieved. The results are a set of predicates that come from different data sources and correspond to subjects, food, objects, and health conditions. We use the notation PRQ_{ml} to represent the set of T predicate results after reasoning the query Q_{ml} for the user u_m . These predicates are a subset of the annotated predicates P in the knowledgebase. We use the notation SRQ_{ml} to represent the set of V subject results after reasoning the query Q_{ml} for the user u_m . These subjects are a subset of the annotated subjects S in the knowledgebase. We use the notation ORQ_{ml} to represent the set of Z object results after reasoning the query Q_{ml} for the user u_m . These objects are a subset of the annotated objects O in the knowledgebase. Below is the model for the retrieved results.

$$PRQ_{ml} = \{p_t \mid p_t \in P, p_t \text{ is given after the process of query } Q_{ml} \text{ for the user } u_m, \\ t=1,2,\dots,T\}$$

$$SRQ_{ml} = \{s_v \mid s_v \in S, s_v \text{ is given after the process of query } Q_{ml} \text{ for the user } u_m, \\ v=1,2,\dots,V\}$$

$$ORQ_{ml} = \{o_z \mid o_z \in O, o_z \text{ is given after the process of query } Q_{ml} \text{ for the user } u_m, \\ z=1,2,\dots,Z\}$$

6.7 Results Conflict Resolution Modeling

The retrieved results can conflict. For example, a conflict happens if we find both statements, "An apple has positive effect on diabetes," and, "An apple can increase the risk of diabetes." We define below a function CF for determining conflicts and a function CFR for resolving the conflicts and a set $NCFP_{ij}$ to hold the un-conflicted predicates.

We use the notation CF to represent a function that determines whether there is a conflict in the retrieved predicates that relate a subject s_i with an object o_i . The function cf_{ij} corresponds to all ordered pairs of (s_i, o_i) from R between a subject s_i and an object o_j . The possible values of the function CF are 0, -1, and +1. The value of the function cf_{ij} is 0 if there is at least one positive predicate that is a member from the set P_{ij}^+ and one negative predicate that is member of the set P_{ij}^- between the subject s_i and the object o_j . The value of the function cf_{ij} equals +1 if all predicates between the subject s_i and the object o_j are positive and are a subset of P_{ij}^+ . The value of the function cf_{ij} equals -1 if all predicates between the subject s_i and the object o_j are negative and are a subset of P_{ij}^- . We show below the model for the function CF .

CF: function to find the conflicts between different predicates

$$CF: R \rightarrow \{0, +1, -1\}$$

$$CF(s_i, o_j) = cf_{ij}$$

$$cf_{ij} = \begin{cases} +1 & , \text{if } P_{ij}^- = \emptyset \\ -1 & , \text{if } P_{ij}^+ = \emptyset \\ 0 & , \text{if } P_{ij}^- \neq \emptyset \text{ and } P_{ij}^+ \neq \emptyset \end{cases}$$

We use the notation RCF to represent a function that resolves any conflict between the predicates that relate a subject s_i with an object o_i if a conflict is found by the function CF .

We define the function rcf_{ij} for all ordered pairs of (s_i, o_i) between a subject s_i and an object o_i from R . The function RCF can have one of three values 0, +1, or -1. The value is based on the value of the function CF and the function WP , which is used for the predicate weight based on the user's interaction as explained in Section 6.3. We use the notation $wp_{m_{ij}}$ to represent the total value of the weighted predicates between the subject s_i with the object o_i that the user interacts with previously. This means that the preferred

predicates from previous user's interactions will take on more weight when it comes to conflict resolution. Below, we show the model of the function $wp_{m_{ij}}$.

$$wp_{m_{ij}} = \sum_{p_{d_{nk}} \in P_{ij}} wp_{m_{p_{d_{nk}}}}$$

As explained in Section 6.3, the sign of the individual values of $wp_{m_{p_{d_{nk}}}}$ is based on the type of predicate (i.e., whether it is positive or negative). The summation of all predicates of a certain subject and a certain object would be either positive if all predicates are positive or negative if all predicates are negative. The summation will be 0 if the positive and negative predicates are equal. In addition to that, the user's interactions are weighted and affect the calculation of the summation by adding more positive or negative strength. If there is a conflict determined by the value 0 of the function cf_{ij} , then the function RCF is used to resolve the conflict in the predicate's subject s_i and object o_i , and we determine the value of the function rcf_{ij} .

The function WP helps in finding the difference between the number of positive and negative predicates. If cf_{ij} equals 0, then rcf_{ij} equals 0 if the number of positive predicates equals the number of negative predicates in P_{ij} . The function rcf_{ij} equals +1 if the number of positive predicates is more than the number of negative predicates in P_{ij} . The function rcf_{ij} equals -1 if the number of negative predicates is more than the number of positive predicates in P_{ij} .

RCF: function for resolving the conflict

$$RCF: R \rightarrow \{0, +1, -1\}$$

$$RCF(s_i, o_j) = rcf_{ij}$$

$$rcf_{ij} = \begin{cases} +1, & \text{if } cf_{ij} = 0, wp_{m_{ij}} > 0 \\ -1, & \text{if } cf_{ij} = 0, wp_{m_{ij}} < 0 \\ 0, & \text{if } cf_{ij} = 0, wp_{m_{ij}} = 0 \end{cases}$$

We use the notation $NCFP_{ij}$ to represent a set of conflict-free predicates between a subject s_i and an object o_i . Next, we explain each function and set in more details. $NCFP_{ij}$ consists of all positive predicates if cf_{ij} or rcf_{ij} equals to +1 while $NCFP_{ij}$ consists of all negative predicates if cf_{ij} or rcf_{ij} equals to -1. $NCFP_{ij}$ is empty set if rcf_{ij} equals to 0 which means there is a conflict that cannot yet be resolved and then we cannot show conflicted results. Below, we show the model for the non-conflicted predicates set $NCFP_{ij}$.

$$NCFP_{ij} = \begin{cases} P_{ij} \cap P_{ij}^+, & \text{if } rcf_{ij} = +1 \text{ or } cf_{ij} = +1 \\ P_{ij} \cap P_{ij}^-, & \text{if } rcf_{ij} = -1 \text{ or } cf_{ij} = -1 \\ \emptyset & , \text{if } rcf_{ij} = 0 \end{cases}$$

6.8 Results Personalization Model

We use the notation Y to represent the set of conflict-free predicates determined in Section 6.7. After the conflicts between the predicates are resolved, the results are personalized based on the user's needs.

First, we calculate the number of occurrences of each predicate in the results within the data sources. We use the notation Y_{ij} to determine the number of predicates between a subject s_i and an object o_i . A higher frequency of certain predicates in a different data source indicates that this information is more trustable. Therefore, we give it more weight.

We use the notation WY to represent a function that determines the weighted occurrence for each predicate. The value of the function WY ranges from 0 to 1, where 0 means that there is no occurrence for the predicate in the results Y and 1 means the predicate has the maximum occurrence. We calculate the value of the WY_{ij} , which determines the occurrence of predicates between a subject s_i and an object o_i , by dividing the number of occurrences of all predicates between s_i and o_j by the maximum number of occurrences of

all predicates in the results, as noted by $Max Y$. For example, we have results of four predicates, the first occurred three times, the second occurred four times, the third is occurred two times, and the fourth occurred one time. So the weight for the first predicate is 3/4, the second predicate is 4/4 (or 1), the third predicate is 2/4 (1/2), and the fourth is 1/4. The model for predicate occurrences is:

$$Y: \{NCFP_{ij}\} \rightarrow \{0, 1, 2, \dots\}$$

$$Y_{ij} = |NCFP_{ij}|, i=1,2,\dots,I, j=1,2,\dots,J$$

$$WY: NCFP \rightarrow [0,1]$$

$$WY_{ij} = \frac{|Y_{ij}|}{Max Y}$$

After finding the occurrences of the predicates, we find the total weight for each predicate. We use the notation WPR_{ij} to represent the function that determines the total weight for each predicate. The value is calculated by adding the preference weight of the same predicate WP_{ij} , the occurrence of the predicate within the results WY_{ij} , the culture preference weight WCP_i , and the preferred elements whether they are in the subject, food, $WVPE_i$, or in the object, health, $WVPE_j$. Finally, we sort the results based on the calculated total weight. Below is the model for predicate weight used in the results personalization.

$$WPR_{ij} = WP_{ij} + WY_{ij} + WCP_i + WVPE_i + WVPE_j$$

CHAPTER 7

HEALTH, FOOD, AND USER'S PROFILE ONTOLOGIES

This chapter introduces the processes used to develop the domain ontologies and the user's profile ontology. The user's profile ontology is based on the user's preferences identified in Chapter 5, and it is integrated with the domain ontologies for semantic manipulation of the user's queries. It is used for query enrichment and results personalization.

The Semantic Web brings the Internet from “web of documents” to “web of data,” where the linked data empower the computers with the ability to provide better services, such as reasoning and inferring. Semantic Web technologies help in building data stores on the web, creating vocabularies, and providing rules to deal with data. Some examples of the technologies used by linked data are resource description framework (RDF), simple protocol and RDF query language (SPARQL), and ontology web language (OWL) (138). *Ontology* is a formal representation of knowledge in a network of concepts within a certain domain using a shared terminology for the types, properties, and relationships between the domain's concepts⁷. The main components of ontologies are:

- concepts: similar to classes in object-oriented programming (OOP);
- instances: similar to objects in OOP;
- attributes: which are part of concept;
- attribute values: which are the values of the attributes and part of the instance;
- subjects: can be concepts, instances, attributes, or attribute values;

⁷ [http://en.wikipedia.org/wiki/Ontology_\(information_science\)](http://en.wikipedia.org/wiki/Ontology_(information_science))

- objects: can be concepts, instances, attributes, or attribute values;
- predicate: relation between a subject and an object; and
- triple: the subject-predicate-object.

RDF is a triple consisting of a subject, predicate, and object. Any SPARQL query comprises a number of triples where the query reasoning engine matches the triples of the SPARQL query with the stored RDF triples in the knowledgebase created during the annotation process. Therefore, a SPARQL query is performed on a RDF dataset that is built based on the annotated web sources.

In (139), the authors presented a methodology to design and develop a Semantic Web search engine to provide accurate results for domain-specific searches. Precise information is retrieved by utilizing the mapping technique between classes and instances. Therefore, the number of search results is reduced along with the search time. The proposed methodology is highly scalable and can fit any domain by providing the required input from the relevant RDF documents to add any domains into the search coverage.

Querying ontologies can take many shapes, and the literature shows too many ways to query ontologies. One category of these is based on graphical user interface (GUI), which provides the ability to navigate and explore an ontology or query an ontology using either templates or a formal ontology query language, such as SPARQL. One famous platform for querying ontology is Protégé, which is used by experts in ontology query language (SPARQL). Another platform eases the semantic search in ontology, such as KIM, and provides a mechanism to query by using predefined templates. As a result, the users will be directed to certain search criteria based on the templates without the need to know the details of the ontology (140).

We explain the development cycle for the multilingual cross-domain ontologies for food, nutrition, and health. First, we define the requirements for each domain. Second, we investigate the existing related ontologies and summarize their limitations with respect to the requirements. Third, we explain how we used the introduced processes to fulfill the requirements. Then we describe the developed ontologies for the food, nutrition, and health domains. Finally, we describe the user's profile ontology and the integration with the domain ontologies.

7.1 Introduction

Ontology is a formal representation of knowledge in a network of concepts within a certain domain using a shared terminology for the types, properties, and relationships between the domain's concepts. Different ontologies are developed for different domains by the domain experts to fulfill certain objectives.

An ontology serves a single domain, while some applications need to use ontologies from different domains to integrate different information sources. Moreover, there could be several ontologies developed for the same domain due to different languages, cultures, expertise, and purposes. Therefore, there is a need to integrate existing ontologies to capture cross-domain knowledge.

As mentioned in the framework chapter, we need to plug domain ontologies into the framework. Ontologies help annotation in having standard references for the acquired knowledge. Thus, web sources can be structured in knowledgebases based on the domain ontologies. These knowledgebases are used in the semantic manipulation of the user's queries to return relevant results.

Our objective is to apply the proposed framework to build a multilingual Semantic Web search application for the food, nutrition, and health domains, as they are critical domains. This will help the community in providing food recommendations based on the user's health conditions. To provide such capabilities, we need integrated ontologies between different domains such as food, nutrition, and health. In addition, we want to use the knowledge discovered in one language to for people using different languages. Such ontologies that satisfy these requirements do not exist. Therefore, we were challenged to develop these ontologies by creating, integrating, and reusing some of the existing ontologies to meet our requirements. Next, we present the processes we have followed in developing these ontologies.

7.2 Ontology Development Processes

There are different methodologies to develop ontologies such as METHONTOLOGY (16), Uschold and King (52), and On-to-Knowledge (47). We introduce four processes below that use some of the existing methodologies. We use these processes to develop multilingual cross-domain ontologies for the food, nutrition, and health domains. The processes are described in the following tables with their inputs, outputs, and possible methodologies that can be followed in each.

TABLE 8 Domain Ontology Development Process

Process no.	1
Process name	Domain ontology development
Description	To develop or reuse certain domain ontology that satisfies the application requirements
Input	Application requirements
Output	Domain ontology
Methodologies	<ul style="list-style-type: none"> - Reuse a single existing domain ontology as is. - Reuse multiple heterogeneous domain ontologies as they are. Some existing methodologies are Fusion and Composition (141). - Extend existing domain ontology. - Build domain ontology from scratch. Some existing methodologies are TOVE, ENTERPRISE, and METHONTOLOGY (141).

TABLE 9 Cross Domain Ontologies Development Process

Process no.	2
Process name	Cross-domain ontologies development
Description	To have integrated cross domain ontologies
Input	Different domains ontologies
Output	Integrated cross-domain ontologies
Methodologies	<ul style="list-style-type: none"> - Reuse an existing integration between different domain ontologies as is. - Extend an existing integration between different domain ontologies (i.e., add additional integration points). - Build an integration between different domain ontologies from scratch (merge ontologies into one ontology, create an integration ontology, and link the ontologies with relationship).

TABLE 10 Multilingual Ontologies Development from Multiple Ontologies Process

Process no.	3
Process name	Multilingual ontologies development from multiple monolingual ontologies
Description	To have integrated multilingual ontologies based on multiple monolingual ontologies
Input	Multiple monolingual domain ontologies
Output	Integrated multilingual domain ontologies using either one-to-one mapping or agnostic ontology acting as a bridge between the existing ontologies
Methodologies	<ul style="list-style-type: none"> - Automatically align the monolingual ontologies (e.g., using translation service, mediator like Wikipedia). - Manually align the monolingual ontologies. - Semi-automatically align the monolingual ontologies (human guided) (i.e., partially automatic and partially manual).

TABLE 11 Multilingual Ontologies Development from Single Ontology Process

Process no.	4
Process name	Multilingual ontologies development from single monolingual ontology
Description	To have integrated multilingual ontologies starting from a single monolingual ontology
Input	Single monolingual domain ontology
Output	Integrated multilingual domain ontologies using either one-to-one mapping or agnostic ontology acting as a bridge between the existing ontologies
Methodologies	<ul style="list-style-type: none"> - Option-1: (create different ontology for each culture) <ul style="list-style-type: none"> o Use a domain ontology development process to create another monolingual domain ontology. o Use multilingual ontologies development from multiple monolingual ontologies process to align the two monolingual domain ontologies. - Option-2: (enrich the existing ontology or replicate it) <ul style="list-style-type: none"> o Automatically translate the input monolingual domain ontology into a new language. o Manually translation the input monolingual

domain ontology.

- Semi-automatically translate the input monolingual domain ontology (human guided).
-

7.3 Ontology Development Cycle

To develop the domain ontologies, the requirements need to be captured from the objective and intended use. Then related existing ontologies are surveyed and assessed as to whether they meet the requirements. Finally, we explain how we follow the introduced processes.

7.3.1 Requirements

We aim to provide answers to questions related to food, nutrition, and health domains. Some examples of these questions are: “Is an apple good for people with heart diseases?” “How much honey can be taken by a diabetes patient?” “What are the health benefits of eating pineapple?” and, “What are the fruits that contain the daily needed quantity of calcium?” To answer such questions, there is a need to have integrated ontologies for different domains: food, nutrient, health (diseases, body parts, body functions), and recipe. Moreover, to answer queries in a different language, the system and ontologies should support a multilingual property. To answer queries that require aggregation of information, we need to have multilevel ontologies. To achieve high relevancy and coverage, we need to use ontologies that have comprehensive and rich vocabularies. To make effective use of the annotation, ontologies’ concept names should be unique and self-contained.

One of the most used and richest knowledgebases for food and nutrition is the U.S. Department of Agriculture (USDA) database. The USDA (18) schema is used as a main guide to develop the core ontology. Foods are gathered into 25 groups according to the USDA classification. There are 146 classes of nutrition. The relation between food and nutrition is based on 100g of food containing a specific amount of nutrition. We will use the USDA as a base for our ontologies selection, and hence we need to assure the alignment possibility for any ontology with the USDA.

7.3.2 Related Ontologies

Based on the criteria discussed above, we have considered related ontologies for food, nutrition, and health. In the next sections, we will present a short description about each one with respect to the requirements given before.

7.3.2.1 Semantic Diet Ontologies

Evan Patton developed a project called Semantic Diet to help people to eat healthier. Patton provided a set of ontologies related to food and nutrition based on the USDA database. We have used these ontologies as a base to build and extend food and nutrition ontologies. Semantic Diet has a main ontology with one concept related to nutrition and two concepts related to food. The two food concepts are based on two USDA food tables: food items and food groups. Semantic Diet has other ontologies: recipes, units for measurements, food serving size, and nutritional guidelines. TABLE 12 shows Semantic Diet ontologies with their corresponding number of instances.

TABLE 12 Semantic Diet Ontologies

Ontology	Number of instances
Recipe	124
Food groups	100
Units for measurements	65
Common measures for foods	118,791
Nutrient	2,847,367
Nutritional guidelines	136

One advantage of Semantic Diet ontologies is that they are built based on the USDA database, which is used in many semantic applications. Another advantage is that they integrate food concepts with nutrition concepts with one property of 100gm.

A disadvantage of Semantic Diet ontologies is that they are flat and shallow ontologies with one to two levels only. This will limit the aggregation at the ontology level. Another limitation is that Semantic Diet ontologies are available in English only. Moreover, many foods contain similar names, which make them difficult to use as-is for annotation. Finally, Semantic Diet ontologies lack synonyms, which leads to limited coverage during annotation.

7.3.2.2 International Classification of Diseases (ICD-10) Ontology

The ICD10 ontology is a formalization in OWL-DL of the International Classification of Diseases (10th ed.), published by the World Health Organization (WHO) in 2004. It is considered a standard tool for health management and other clinical purposes and is utilized to track the occurrence and frequency of diseases and illnesses⁸.

ICD10 is huge ontology consisting of 14,502 concepts of diseases and health care procedures, which provides a huge vocabulary set. Although the ontology is available in the English language, the translations are available for the vocabularies of ICD10 in different languages such as Arabic. Having a wide vocabulary set and availability of vocabularies in different languages is a positive aspect of the ontology.

The ontology is designed to categorize diseases and health issues based on the various types of health and important records. The ontology is hierarchical in nature and classifies all these concepts into many levels, such that the concepts are not self-explanatory unless a complete parent hierarchy is observed to understand the actual vocabulary for a set of concepts. Moreover, the ontology uses the technical names of diseases and does not have embedded synonyms. Such ontology makes the text processing less effective as more work is required to map the ontology concepts to the text being annotated.

7.3.2.3 Human Disease Ontology

The Disease Ontology (DO) is an open-source ontology for the integration of biomedical data associated with human disease. Terms in DO ontology are well defined and use standard references. These terms are linked to well-established, well-adopted

⁸ <http://www.who.int/classifications/icd/en/>

terminologies that contain disease and disease-related concepts such as SNOMED, ICD-9 and ICD-10, MeSH, and UMLS.

DO ontology represents a comprehensive knowledge base of 8043 inherited, developmental and acquired human diseases. Each concept has a reference for most common health related ontologies with different synonyms or alternative names for the same concept. It is very useful for semantic annotation for two reasons: self-contained names used for each concept and a rich set of synonyms for each concept. For those reasons, we have selected this ontology for our test case for semantic annotation of disease concepts.

The only limitation of the DO ontology is related to multilingual support since it is only provided with English names only.

7.3.2.4 AGROVOC Ontologies

AGROVOC provides ontologies with a rich vocabulary that covers different areas of the Food and Agriculture Organization (FAO) of the UN, such as food and nutrition. It contains more than 32,000 concepts and supports different languages such as Arabic, Chinese, Czech, English, French, German, Hindi, Hungarian, Italian, Japanese, Korean, Lao, Persian, Polish, Portuguese, Russian, Slovak, Spanish, Thai, and Turkish. AGROVOC uses the standard RDF format to represent their linked dataset.

The main advantage of the AGROVOC is the multilingual support that includes 22 languages with four languages under development. The second advantage is the way AGROVOC organizes its concepts in a hierarchy supporting multilevel concepts. Another advantage for AGROVOC is its support of synonyms in different ways for any concept within the ontologies.

One of the major limitations of AGROVOC ontologies is the lack of USDA alignment, which limits its integration with other ontologies that have relations with other domains.

7.3.2.5 FOODS Ontology

The FOODS ontology contains nine main concepts: regional cuisine, dishes, ingredients, availability, nutrients, nutrition-based diseases, preparation methods, utensils, and price. Food concepts are divided into three main categories: beverages, plant-based food, and animal-based food. Each food category contains deeper levels of concepts. The nutrition concept is categorized into six concepts: carbohydrates, proteins, vitamins, fats, minerals, and water. The ontologies are used as part of a system to answer questions raised by users based on their profiles and relationships between the ontology concepts.

The advantages of the FOODS ontology are the integration between food, nutrition, and disease. The ontology is a very useful for annotation since it was built for computer processing.

The FOODS ontology contains a shallow hierarchy of concepts with two or three levels only. The ontology contains only an English version with no synonym. It is not aligned with the USDA food database. With that limitation, we did not consider this ontology as a basis for our system.

7.3.2.6 PIPS Ontologies

The Personalized Information Platform for Life & Health Service (PIPS) (141) provides multiple ontologies related to food, nutrition, clinical record, recipe, menu, and person. These ontologies contain 261 concepts and two object properties. Foods are gathered into 13 groups: vegetables, fruits, grain products, special nutrition, beverages, sea foods, egg products, oils and fats products, meat, soups and sauces, sugar products, nuts and seeds,

and milk products. The nutrition ontology contains a different grouping of foods based on the different use of food.

TABLE 13 PIPS Ontologies

Ontology	Number of concepts	Object properties
Food	180	2
Nutrition	18	3
Recipe	8	4
Menu	7	3
Clinical record	14	4
Profile	3	4

The advantages of PIPS ontologies are the integration between provided ontologies. Moreover, the naming of concepts is usable for animations. The PIPS ontologies are provided with English labels only and are without synonyms. Ontology resources are not linked to external entities such as the USDA food database. The hierarchy levels of the ontologies concepts are shallow.

7.3.3 Comparison and Limitations

To decide which of those ontologies we should use, we developed a comparison table based on hard and soft criteria. The first hard criterion for food and nutrition is the alignment to the USDA food and nutrition database. The second hard criterion is the usability of those ontologies for annotation with respect to naming schema and uniqueness

for concept labels. Other criteria are considered soft and can be handled using the approaches described before.

TABLE 14 Food and Nutrition Ontologies

Ontologies	USDA alignment	Multi-level	Richness	Multilingual	Usability
Semantic	Yes	Low	Low	No	Low
Diet					
PIPS	No	Low	Low	No	High
FOODS	No	High	Low	No	High
AGROVOC	No	High	Low	Yes	High

TABLE 15 Health Ontologies

Ontology	Link to other references	Multi-Level	Richness	Multilingual	Usability
ICD-10	no	med	low	limited	low
Human disease	Yes	High	High	No	High

7.3.4 Ontology Development Cycle to Fulfill the Requirements

In this section, we show what ontologies were used from the above choices and why. We show what modifications were done and how by explaining the steps as per the approaches already discussed. We show also that some ontologies had to be created because we did not find existing ontologies that fulfilled the requirements.

7.3.4.1 Food Ontology.

Based on the hard rules, we have selected the Semantic Diet food ontology only because it provides the two hard rules of being aligned with the USDA food database and being useful for annotation. The limitation for this ontology is the hierarchy levels and lack of a multilingual property. For hierarchy levels, we have extended the ontology with four to five levels, as needed, in addition to the two levels provided by the initial Semantic Diet ontology. The multilingual property is achieved by adapting process number 3 to produce a multilingual ontology that covers English and Arabic languages at this stage. We maintain the same integration with nutrition concepts.

7.3.4.2 Nutrition Ontology

Similar to food, we have selected nutrition ontology provided by Semantic Diet as an initial ontology. The Semantic Diet nutrition ontology contains only one concept with 146 distinct nutrition elements with instances for all food instances. We have extended the ontology to many levels to capture the aggregation of nutrients in the same group. The multilingual property is achieved by adapting the process number 3 to produce a multilingual ontology that covers English and Arabic languages at this stage. We maintain the same integration with food concepts.

7.3.4.3 Recipe Ontology

Similar to food and nutrition, we have selected recipe ontology provided by Semantic Diet as an initial ontology. The Semantic Diet recipe ontology contains only one concept without any instances. We have extended the ontology to many levels to capture the aggregation of recipes in the same group. The multilingual property is achieved by adapting the process number 3 to produce a multilingual ontology that covers English and Arabic languages at this stage. We maintain the same integration with food concepts.

7.3.4.4 Disease Ontology

We have adapted the human disease ontology (DO) because it is the most useful of our choices for annotation. The multilingual property is achieved by adapting the process number 3 to produce a multilingual ontology that covers English and Arabic languages at this stage. We defined different interaction with food and nutrition concepts.

7.3.4.5 Body part and body function ontologies

Since we did not find suitable ontologies that cover concepts related to the human body, either functions, systems, or parts, we used some information about the human body and built a primitive ontology to cover those two concepts.

7.4 Health and Food Ontologies Description

We went through different ontologies throughout the development cycle of the whole framework implementation. We chose different ontologies and then gave precedence to one or the other based on different facts that directly affect the effectiveness of the

process. Mostly text processing is the area that drove the focus on keeping the best ontology in terms of the right vocabulary and more search space for text mapping.

7.4.1 Disease Ontology

The ontology for disease is the human disease ontology. We choose this ontology because its concepts are self-contained concepts, unlike the ICD10 (WHO). Having self-contained concepts is more suitable when text processing as the concept is independent of the parent concepts and is meaningful enough to map to the contextual words during the text processing. In general, the human disease ontology is a comprehensive vocabulary that is hierarchical in structure. For the description of ontologies in terms of metrics, it has 8,685 concepts. It holds 15 properties, and the maximum depth of the concepts is 14. On average, there are three child concepts for each concept, while the maximum number of child concepts is 80.

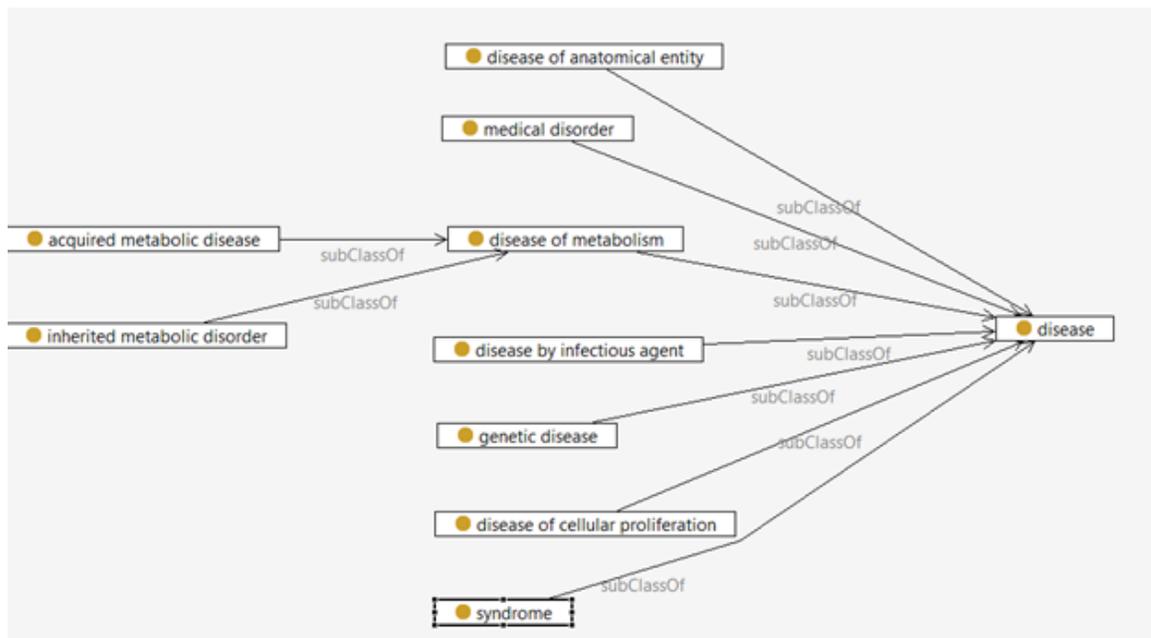


Figure 26 Disease Ontology

7.4.2 Food Ontology

The ontology for food is adapted from Semantic Diet as they based their ontology on the USDA database for food items and classifications of the food groups. The ontology is available in English, so we added the translation of the ontology into Arabic to have a test case of multilingual support. This ontology is just one main concept of FoodItem and all the food items instances belonging to it, which are 9,000. The classification of FoodItem is handled through FoodGroup concept.

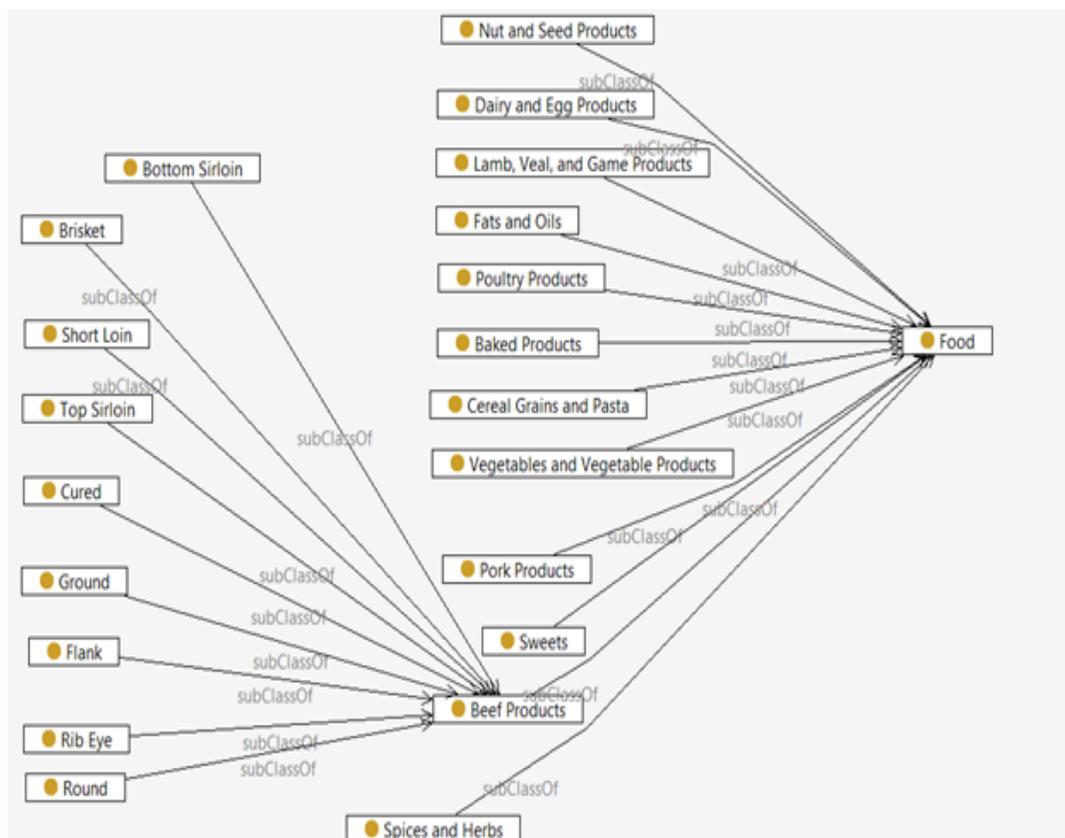


Figure 27 Food Ontology

- صحة
 - عضو بدني
 - مرض
 - وظيفة بدنية
- علاقة
 - مادة غذائية
- ▲ مجموعة غذائية
 - أغذية أطفال
 - الأسماك والمحار المنتجات
 - الأطعمة السريعة
 - الأطعمة العرقية
 - الألبان ومنتجات البيض
 - البقول ومنتجات البقوليات
 - التوابل والأعشاب
 - الجوز والبذور المنتجات
 - الحبوب والمعكرونة
 - الحساء والصلصات، والمرق
 - الحلويات
 - الخضروات والمنتجات النباتية
 - الدهون والزيوت
 - الضأن، لحم العجل، ومنتجات لعبة
 - الفواكه وعصائر الفواكه
 - النقانق واللحوم غداء
 - خبز المنتجات
 - فطور من الحبوب
 - لحوم البقر المنتجات
 - مشروبات
 - مطعم للأغذية
 - منتجات الدواجن
 - منتجات لحوم الخنزير
 - وجبات، مقبلات، وأطباق جانبية
 - وجبات خفيفة

Figure 28 Arabic Food Ontology

7.4.3 Nutrition Ontology

The nutrition ontology is taken from Semantic Diet and is based on the USDA database. Since the USDA database hold information about the nutrients for food items, the ontology acquires the same relations between food ontology. Similarly, we provided the Arabic translation for nutrients.

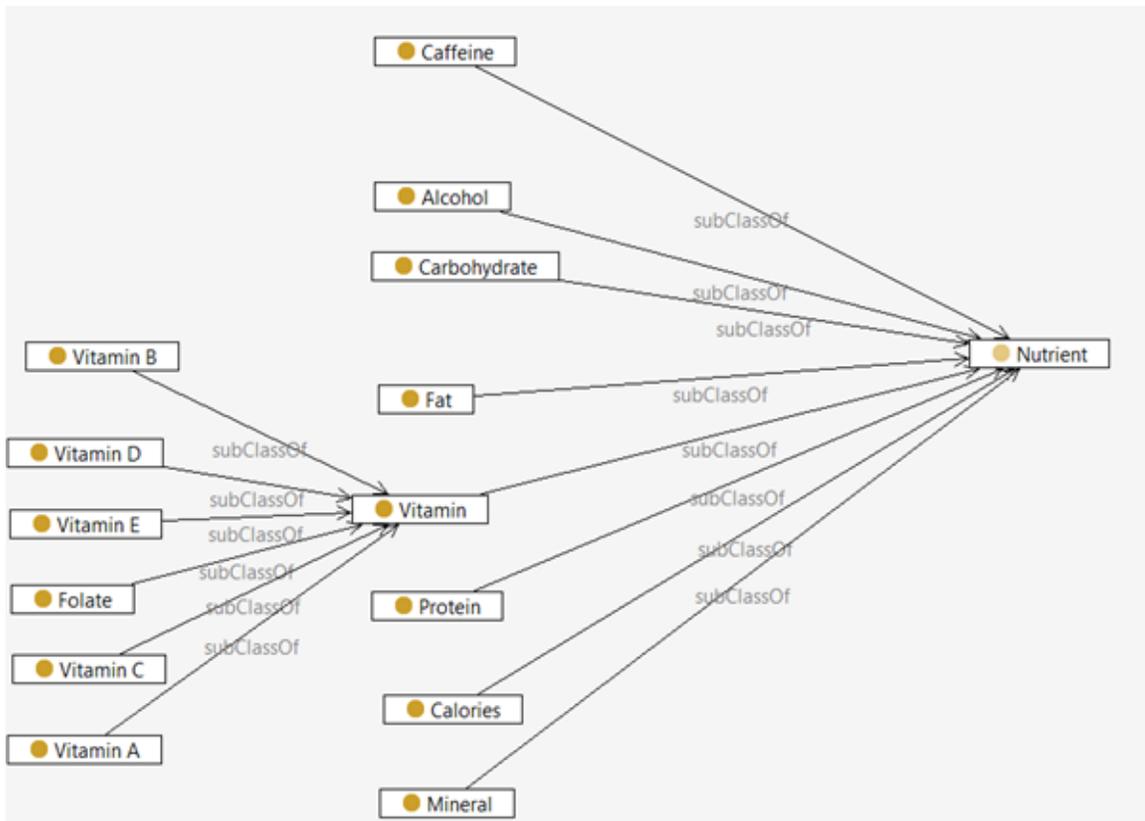


Figure 29 Nutrition Ontology

7.4.4 Body Function and Body Part Ontologies

These are small, self-created ontologies for the proof of the concept. Any available ontology could be adopted, but, unfortunately, no comprehensive ontology was available for body functions or body parts. These are small ontologies with 60 instances for body functions and 163 for body parts.

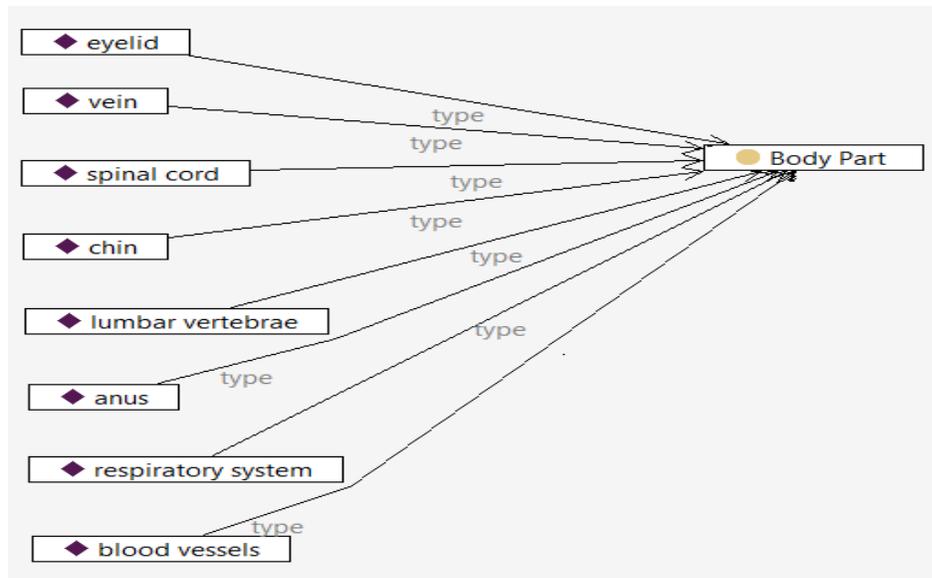


Figure 30 Body Part Ontology

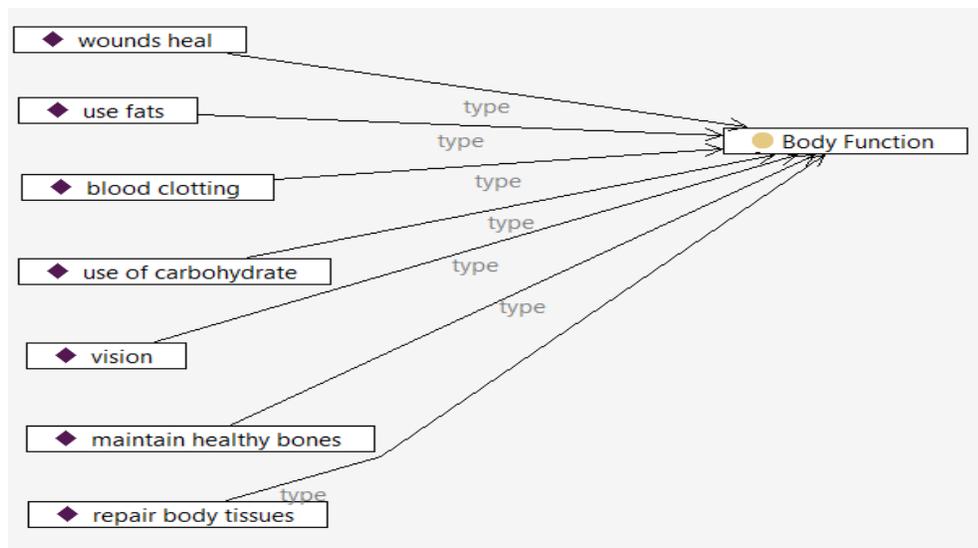


Figure 31 Body Function Ontology

7.4.5 Integration Ontology

The integration ontology is the upper layer ontology that integrates the health ontologies (disease, body parts, body functions) with food- (food item and nutrient) related ontologies. It is done through using the commonly known relations among the domains, which will allow us to capture and reason information following the used relations.

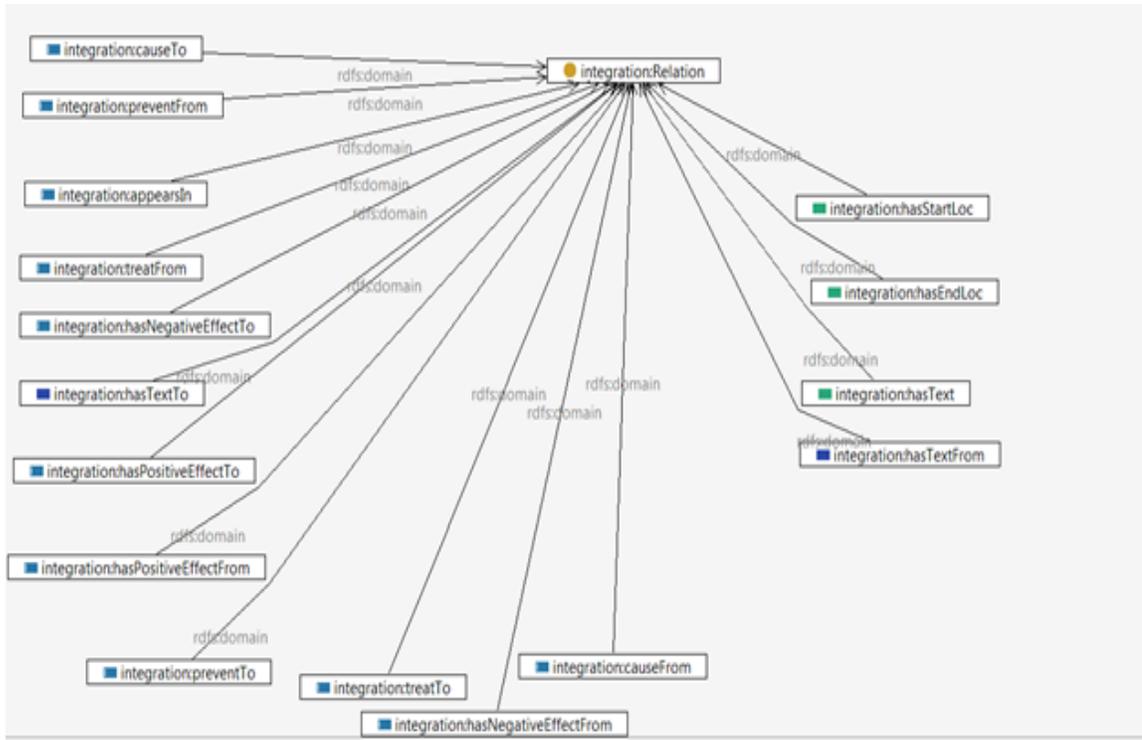


Figure 32 Integration Ontology

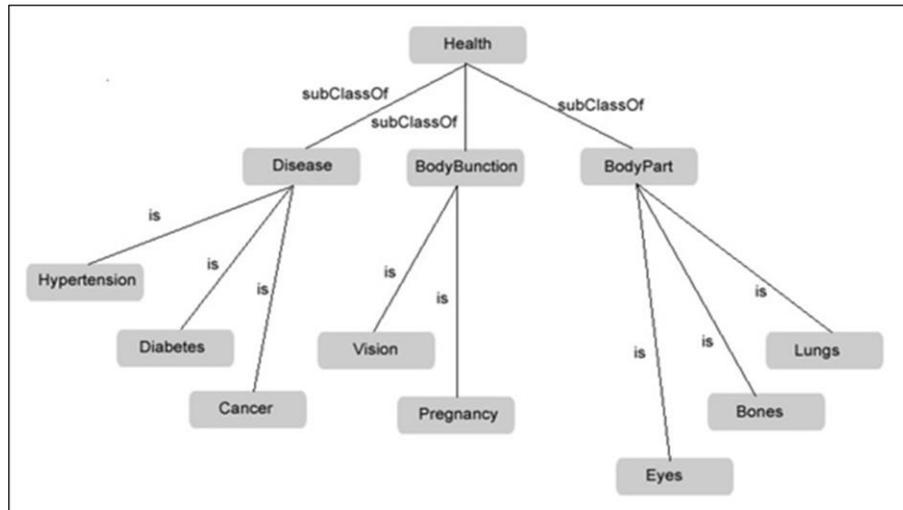


Figure 33 Integrated Health Ontology

TABLE 16 Summary of Developed Ontologies

Ontology	Number of Concepts	Number of Instances
Food	1	8648
Nutrition	182	225
Recipe	1	806
Disease	7277	5491
Body part	1	163
Body function	1	60
Profile	1	0
Integration	4	NA
USDAFood (for reference)	1	15100
FoodGroup	613	15100

7.5 User's Profile Ontologies

The user's profile represents what the user likes and dislikes. It is needed for personalizing the recommendations. It can be represented in different ways, such as in a keywords profile, which assigns the keyword with weight based on the user's preferences. More details of different ways are found in (10). We have selected representing the profile as an ontology because this work is related to other parts in a bigger project, which are based on semantic ontologies. Representing the profile as ontology makes it easier to integrate with the health and nutrition domains' ontologies and helps in reasoning the information using semantic languages such as SPARQL.

Initially, we collect the user profile, which contains the food preference, health conditions, culture, and economic status, using a form. The profile is updated by an analysis of the interactions of the user with the results, which improves the future results. For example, when the user always selects a specific food from the results the profile is updated to show that the user likes this specific food. We use the ontology concept to represent the profile and for use in the semantic search. Figure 36 shows the details of the user's profile ontology and divides it into four parts. The first part contains the basic information of the user such as name and gender. The second part contains the basic health information of the user such as blood type and body mass index (BMI). The third part contains the medical record information of the user such as diseases and whether the user is pregnant. The fourth part contains the user's usage statistics information such as older searches and visited links. The relationship between two concepts is shown as a dashed arrow and that refers to the "triple" quality in RDF terms (12).

7.5.1 Religion Ontology

We had a need to create religion ontology to map the profile, health, and food ontologies to the related religion properties. The religion ontology depends on the other developed domain ontologies and contains properties shared and relations with these ontologies. Hence, we create a religion ontology as a new ontology to answer questions related to food preference with regard to the user's religion. Figure 34 shows the religion ontology.

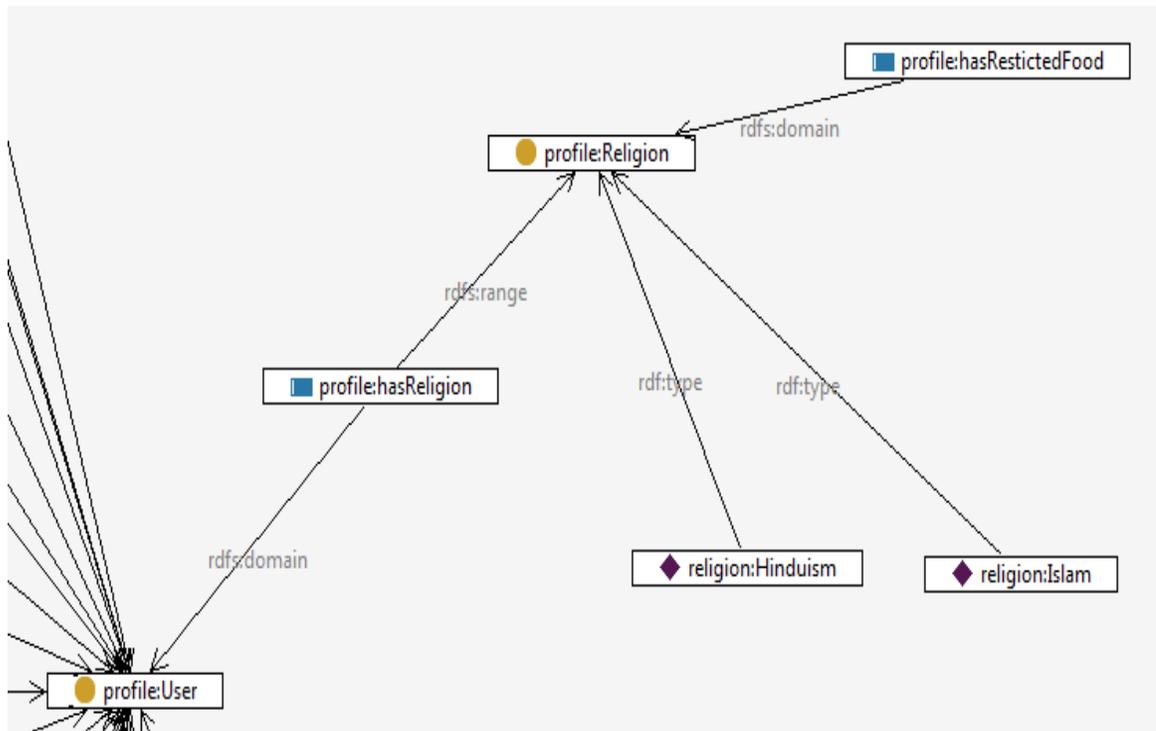


Figure 34 Religion Ontology

7.5.2 Culture Ontology

We had a need to create a culture ontology to map the profile, health, and food ontologies to the related culture properties. The culture ontology depends on the other developed domain ontologies and contains properties shared and relations with these ontologies. Hence, we create a culture ontology as a new ontology to answer questions related to food preference with regard to the user's culture. Figure 35 shows the culture ontology.

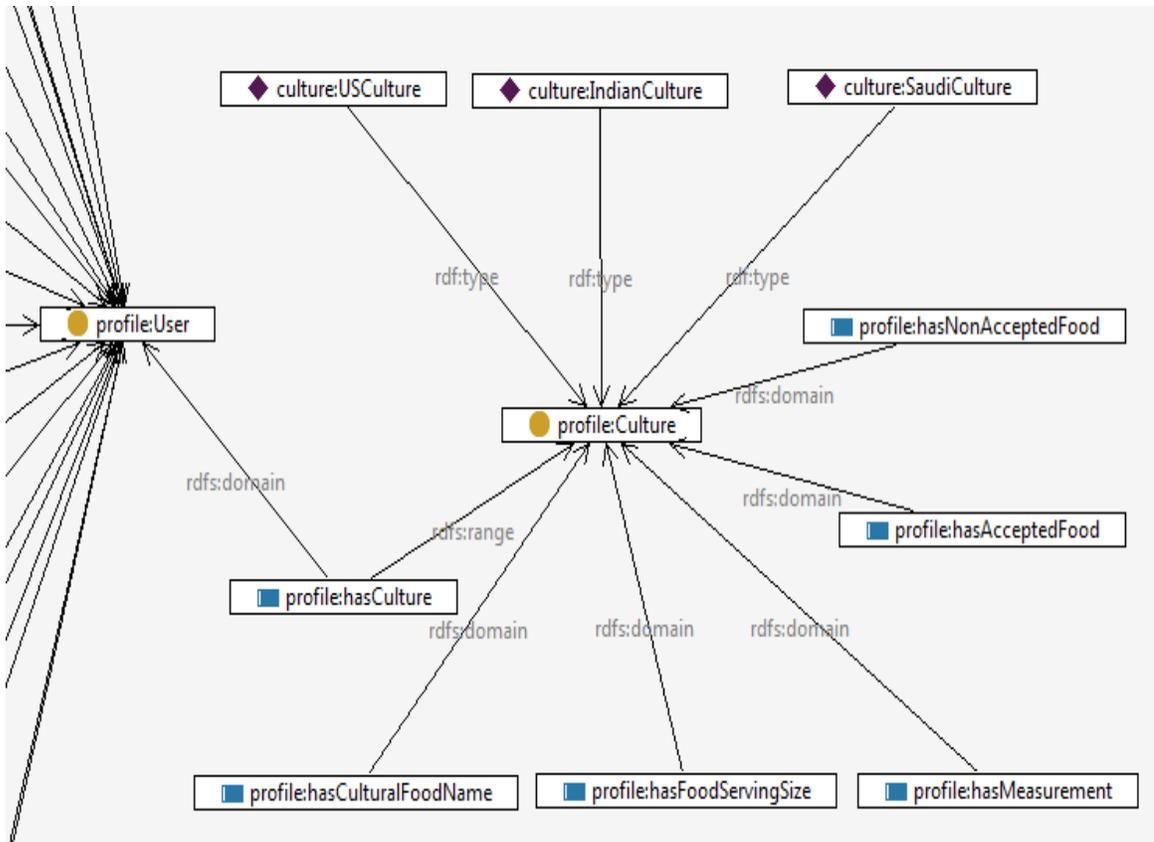


Figure 35 Culture Ontology

7.5.3 User's Profile Ontology

We have created a new profile ontology based on food and health ontologies. The mix between personal information and specialized food and health information motivates creating a specific profile ontology that can help in personalizing the food and health information. We did not find a suitable ontology that covers both sides. The ontology is linked with disease ontology, body part ontology, body function ontology, food ontology, nutrition ontology, and recipe ontology. Figure 36 shows the user's profile ontology.

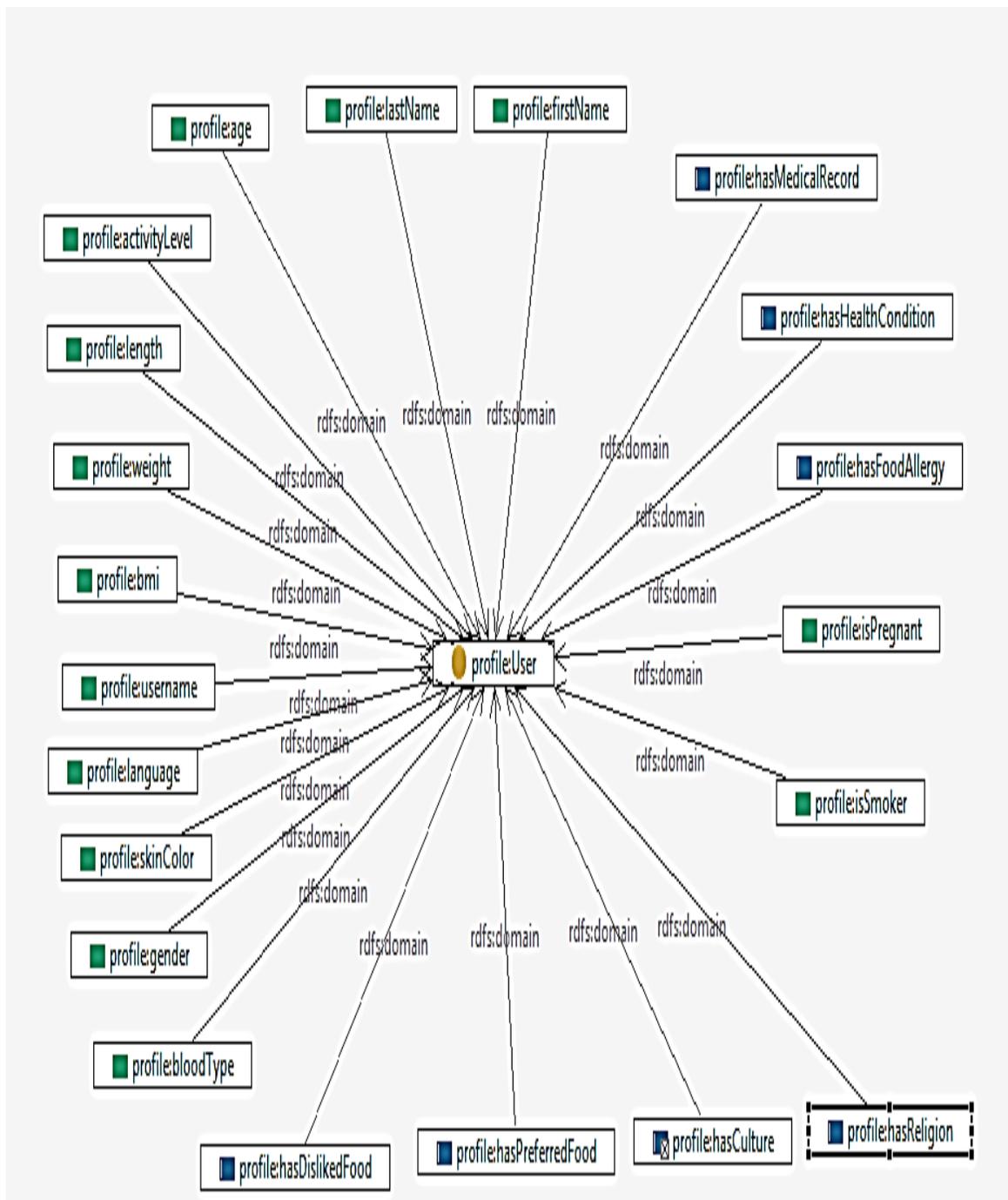


Figure 36 User's Profile Ontology

CHAPTER 8

IMPLEMENTATION: HEALTH AND FOOD DOMAIN

CASE STUDY

In this chapter, we show the details of the development of the proposed framework on health and food domains. We start with a motivation scenario, then a requirements analysis, the design, and then the implementation details.

8.1 Motivation Scenario

To realize the importance of implementing the proposed framework, we show a motivation scenario. In this scenario, Ali is 40-year-old patient with diabetes. In the culture Ali lives in, he is invited to Iftar, dinner, in Ramadan, the month where Muslims fast during the day and eat after sunset. He needs health advice and to know what food he should take during these dinners. He will pass on this advice to his relatives and friends to have them prepare the appropriate food for him when he is goes to the dinner. He opens the ASPIR SYSTEM using his iPad, where he has a profile with some basic information. He checks his nutrition requirements based on his health and medical information to guide his relatives when they prepare his meals. He types in ASPIR SYSTEM, “What food is suitable for me?” ASPIR SYSTEM recognizes that his profile does not have enough information about Ali. ASPIR SYSTEM asks Ali some questions to narrow down his question and give him the right advice. It asks him about his weight; Ali answers with

100kg. It asks him about his height; Ali answers 170 centimeters. It asks him about his blood type; Ali answers with A+. It asks Ali if he has an electronic medical record; Ali enters some identification information to retrieve his medical record and medical history. ASPIR SYSTEM finds out that he has diabetes. After getting enough information, ASPIR SYSTEM gives Ali healthy advice for his food. It starts analyzing his query to generate an annotated query that is ready for reasoning. The query is then enriched with the profile information. Then ASPIR SYSTEM searches the knowledgebase for relevant results and personalizes the results based on Ali's profile. Ali interacts with the results, and ASPIR SYSTEM monitors his interactions to refine his profile.

8.2 Requirements Analysis

We have analyzed the requirements for the ASPIR framework to design and then implement the framework. The main requirements of the framework are:

- user submits query and gets semantic personalized results;
- user creates and manages the personal profile;
- user accesses the system from different platforms (mobile, desktop);
- user gets results from trusted sources;
- user's feedback on the results is captured (explicitly or implicitly); and
- system supports multilingual queries and provides results based on query's language.

This addresses the needs of reliable, semantically integrated, and personalized health and nutrition information with multilingual and cultural support. TABLE 17 shows the benefits and supporting features of the developed system.

TABLE 17 Benefits and Supporting Features

Benefits	Supporting Features
A user wants to get trusted health and food information.	Find trusted health and food information.
A user wants recommendations based on his or her profile.	Personalize search results and recommendation.
A user can write queries in any understood way without being restricted to certain keywords.	Semantically processed user queries.
A user can get results from different domains related to any query.	Provide results from multiple domains related to health, food, nutrition, diseases, etc.
A user can search in any language and any culture.	Support different languages and different cultures.

8.2.1 Use Cases and Sequence Diagrams

Figure 37 shows the use case diagram based on the collected diagrams. It contains two actors, the end user and the translator, and six use cases: register, manage profile, search, navigate results, use user's profile, and provide feedback. It includes some use cases from the other two components mentioned in Chapter 3.

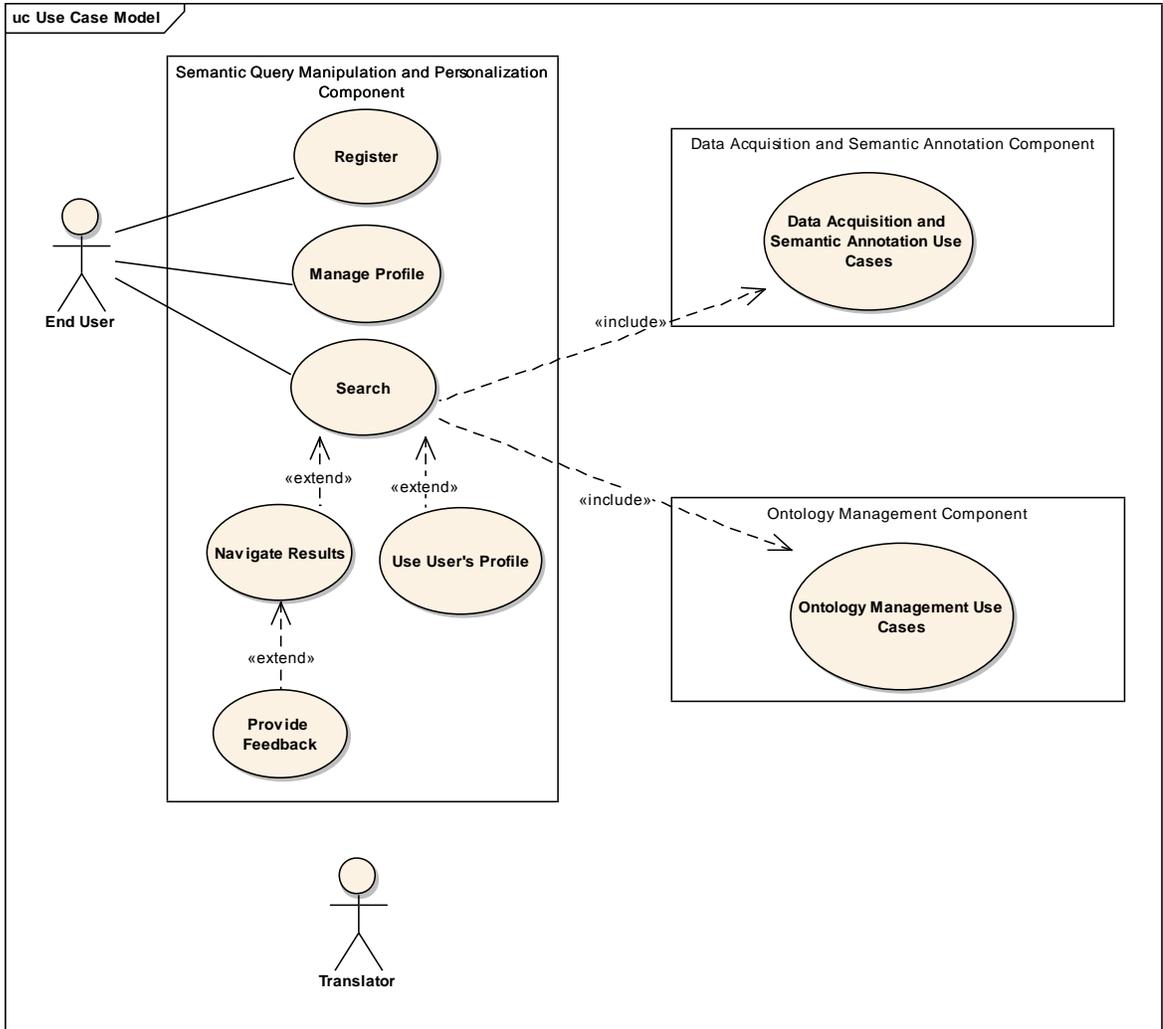


Figure 37 Use Case Diagram

Next, we explain each use case in a separate sub-section.

8.2.1.1 “Search” Use Case

TABLE 18 shows the specification of the “search” use case. It describes the use case showing the preconditions, post-conditions, and other related information.

TABLE 18 “Search” Use Case Specification

Description:	This use case performs search on the knowledgebase based on the query and displays the matching result.
Preconditions:	The user has logged in into the system; otherwise the user is considered as anonymous.
Post conditions:	The results of the user’s query are presented, if any; otherwise it shows no result.
Frequency of use:	High.
Actor(s)	End user, Translator.
Normal course of events:	<ol style="list-style-type: none"> 1. User enters the query in the query box. 2. System annotates the query. 3. System gets the user’s profile. 4. System enriches the query with the user’s profile. 5. System matches the annotated query with the corresponding query template. 6. System executes the matched query template. 7. System refines the result based on the user’s profile. 8. System displays the result on the user’s screen.
Alternative courses:	None.
Extends:	None.

Exceptions: No result is found. Message is shown to the user.

Includes: Navigate result: Use user profile.

Figure 38 shows the sequence diagram of the “search” use case and its interactions with other use cases and objects.

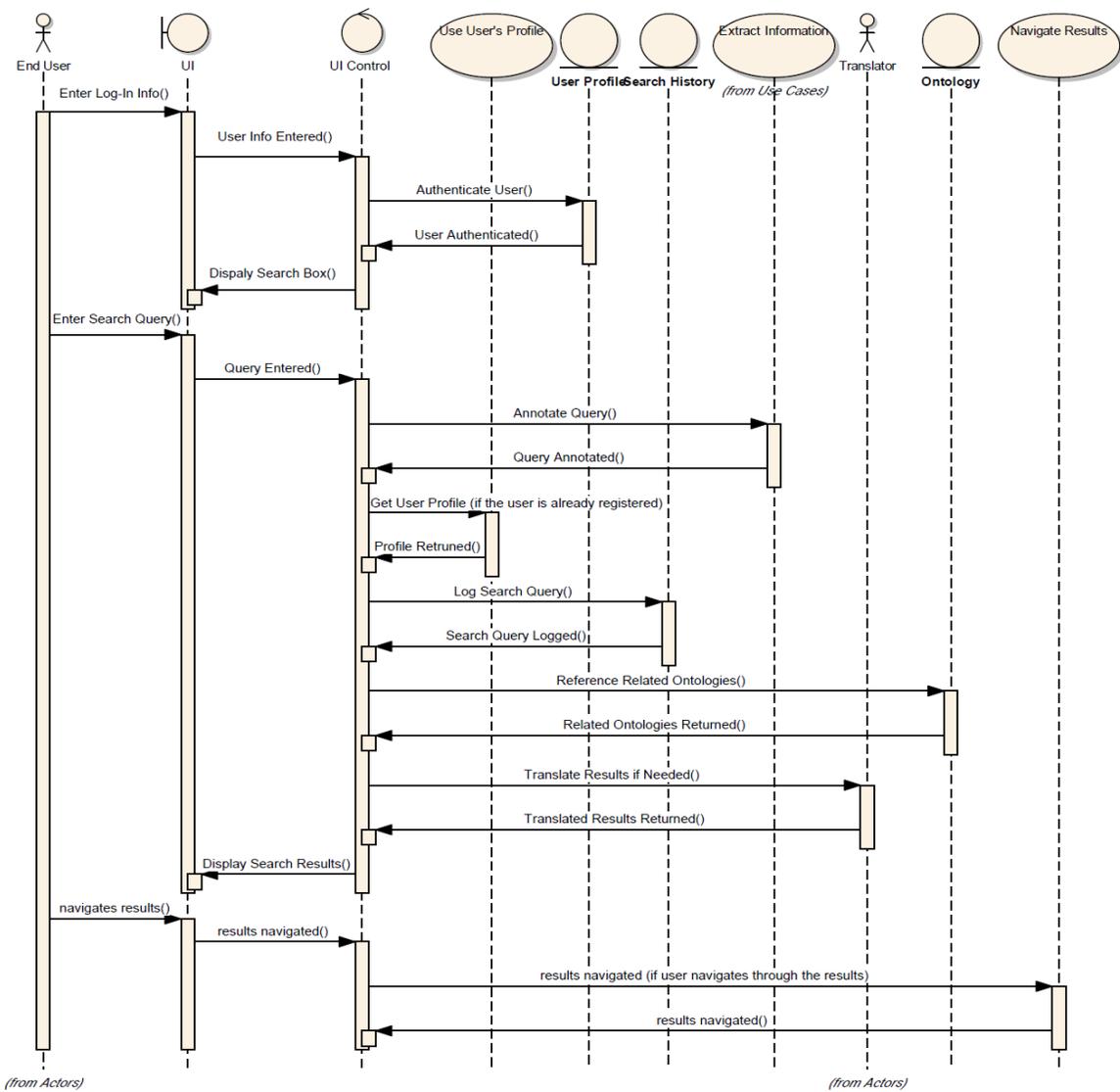


Figure 38 Sequence Diagram for “Search” Use Case

8.2.1.2 “Register” Use Case

TABLE 19 shows the specification of the “register” use case. It describes the use case showing the preconditions, post-conditions, and other related information.

TABLE 19 “Register” Use Case Specification

Description:	This use case allows the user to register in the system and creates a user’s profile with the entered values.
Preconditions:	User selects the registration screen.
Post conditions:	A user’s profile is created with the corresponding values entered by the user.
Frequency of use:	Medium.
Actor(s)	End User.
Normal course of events:	<ol style="list-style-type: none">1. User enters the information in the registration form.2. System verifies the user’s input.3. System creates the user’s profile and notifies the user.
Alternative courses:	None.
Extends:	None.
Exceptions:	User already defined. Duplicate profile for the same user is not allowed. Message is shown to the user.

Includes: None.

Figure 39 shows the sequence diagram of the “register” use case and its interactions with other use cases and objects.

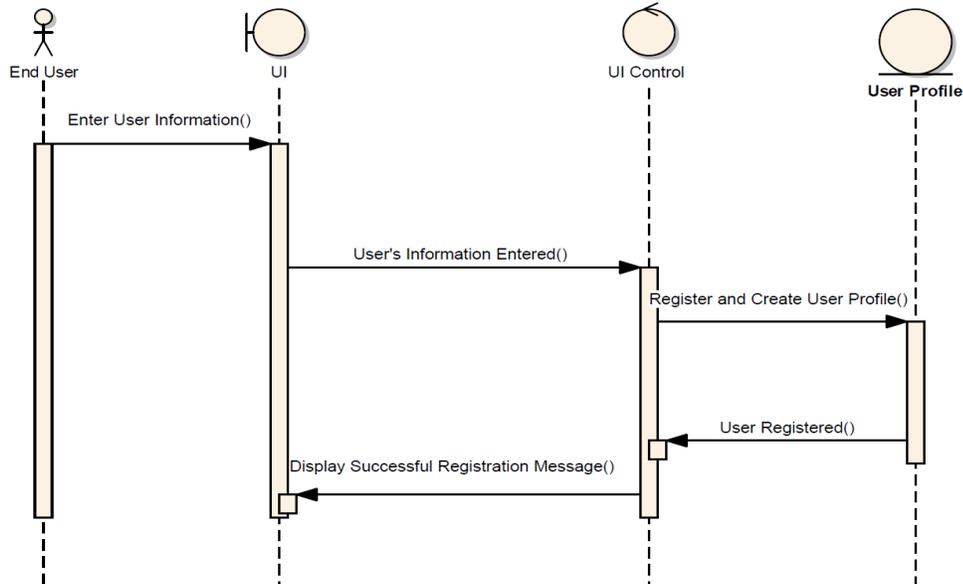


Figure 39 Sequence Diagram for “Register” Use Case

8.2.1.3 “Manage User’s Profile” Use Case

TABLE 20 shows the specification of the “manage user’s profile” use case. It describes the use case showing the preconditions, post-conditions, and other related information.

TABLE 20 “Manage User’s Profile” Use Case Specification

Description:	This use case allows the user to update his or her profile.
Preconditions:	User already is logged in.
Post conditions:	User profile is updated according to the user’s entered values.
Frequency of use:	Medium.
Actor(s)	End user.
Normal course of events:	<ol style="list-style-type: none"> 1. User enters the information in the profile’s screen. 2. System verifies the user’s input. 3. System updates the user’s profile and notifies the user.
Alternative courses:	None.
Extends:	None.
Exceptions:	None.
Includes:	None.

Figure 40 shows the sequence diagram of the “Manage user’s profile” use case and its interactions with other use cases and objects.

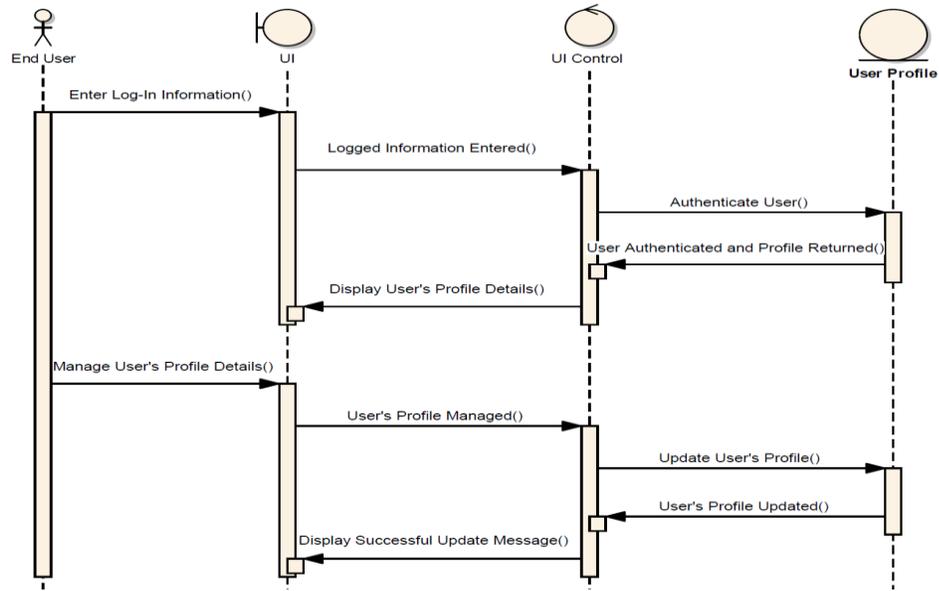


Figure 40 Sequence Diagram for “Manage User’s Profile” Use Case

8.2.1.4 “Navigate Results” Use Case

TABLE 21 shows the specification of the “navigate results” use case. It describes the use case showing the preconditions, post-conditions, and other related information.

TABLE 21 “Navigate Results” Use Case Specification

Description:	This use case allows the user to navigate through the retrieved results.
Preconditions:	Search result is displayed on the screen.
Post conditions:	User’s profile is updated according to the user’s navigation.
Frequency of use:	High
Actor(s)	End user.
Normal course of events:	<ol style="list-style-type: none"> 1. User reacts with the results. 2. System adds the user activities in the activity log.
Alternative courses:	<ol style="list-style-type: none"> 1. User enters explicit feedback on the search result. 2. System stores user feedback using “provide feedback” use case.
Extends:	Search.
Exceptions:	None.
Includes:	Provide feedback.

Figure 41 shows the sequence diagram of the “navigate results” use case and its interactions with other use cases and objects.

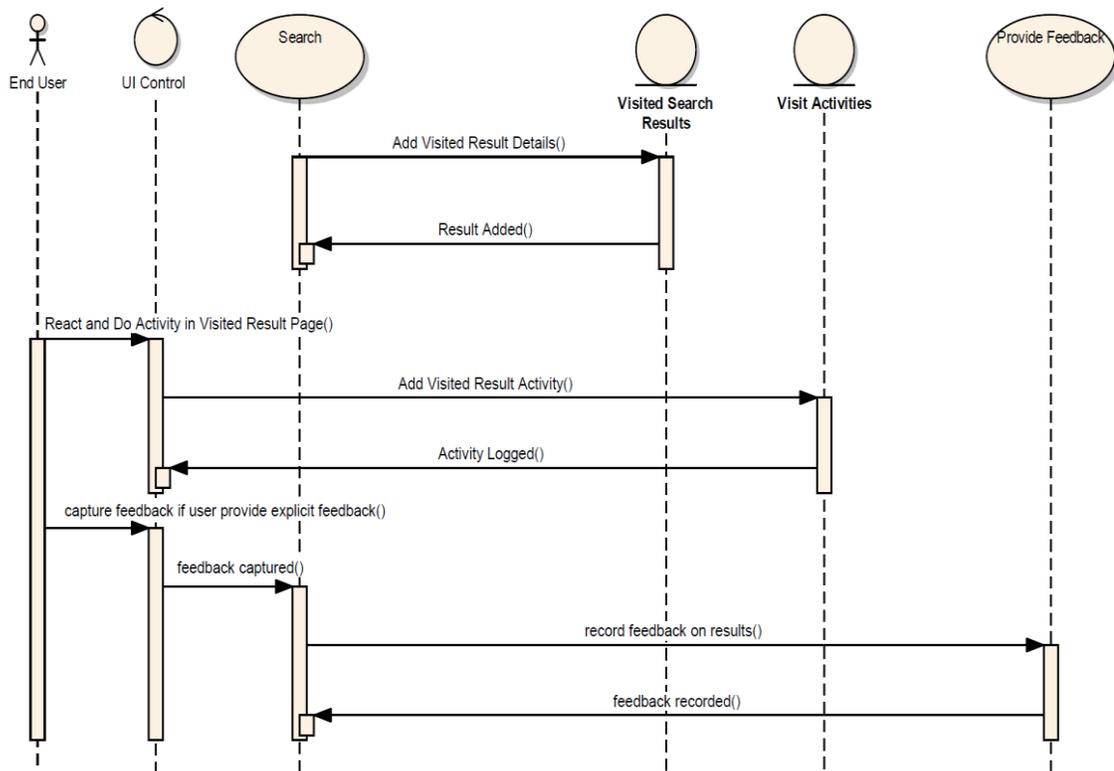


Figure 41 Sequence Diagram for “Navigate Results” Use Case

8.2.1.5 “Use User’s Profile” Use Case

TABLE 22 shows the specification of the “use user’s profile” use case. It describes the use case showing the preconditions, post-conditions, and other related information.

TABLE 22 “Use User’s Profile” Use Case Specification

Description:	This use case fetches the user’s profile from persistent store and returns it into the preferred format.
Preconditions:	User is logged in.
Post conditions:	The requested user profile is returned according to the user’s requested format.
Frequency of use:	High.
Actor(s)	None.
Normal course of events:	<ol style="list-style-type: none"> 1. The requested user’s profile comes from “search” use case. 2. System fetches the user’s profile from the persistent store. 3. System transforms the user’s profile as per the requested format. 4. System returns the user’s profile to “search” use case.
Alternative courses:	None.
Extends:	Search.
Exceptions:	The requested user’s profile does not exist or the requested format is invalid. Message is shown to the

user.

Includes: None.

Figure 42 shows the sequence diagram of the “use user’s profile” use case and its interactions with other use cases and objects.

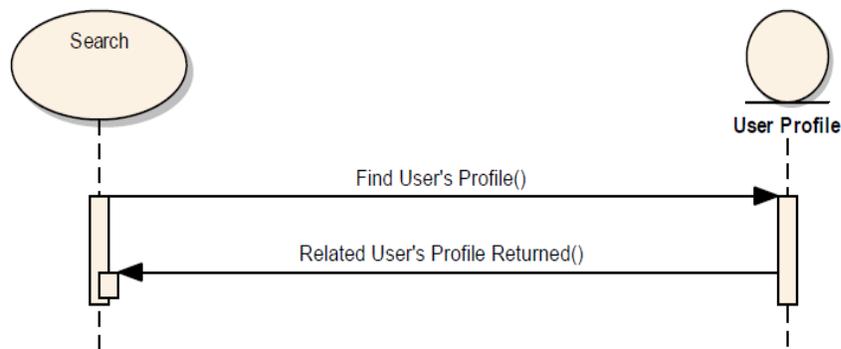


Figure 42 Sequence Diagram for “Use User’s Profile” Use Case

8.2.1.6 “Provide Feedback” Use Case

TABLE 23 shows the specification of the “provide feedback” use case. It describes the use case showing the preconditions, post-conditions, and other related information.

TABLE 23 “Provide Feedback” Use Case Specification

Description: This use case stores the explicit user feedback during the navigation on search results.

Preconditions: User is logged in.

Post conditions: The explicit feedback is stored and the user’s profile is updated.

Frequency of use: Low.

Actor(s) None.

Normal course of events: User gives explicit feedback during search “result navigation” use case.

Alternative courses: “Navigate result” use case sends the feedback to be stored.

Extends: System stores the feedback into persistent store.

Exceptions: None.

Includes: Navigate result.

Figure 43 shows the sequence diagram of the “provide feedback” use case and its interactions with other use cases and objects.

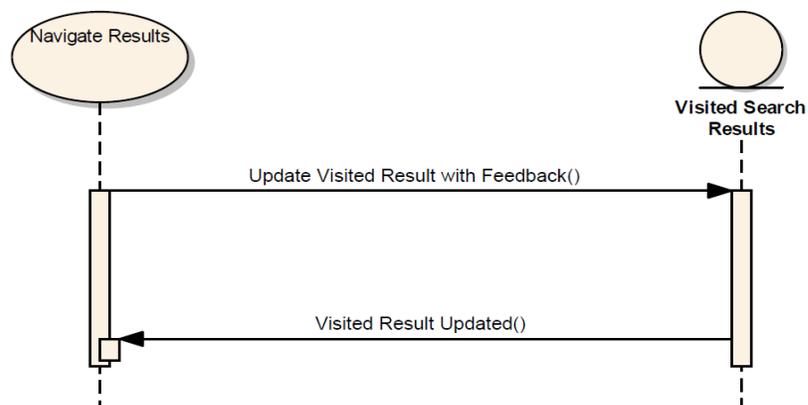


Figure 43 Sequence Diagram for “Provide Feedback” Use Case

8.3 Design

We use the model-view-controller (MVC)⁹ design pattern for the web part of the system. The user's interactions are handled by the controller, which dispatches the requests to different models. Each model corresponds to a Java server page (JSP), which renders the webpage and shows it to the user. Figure 44 presents the MVC model used to handle the user's Web requests.

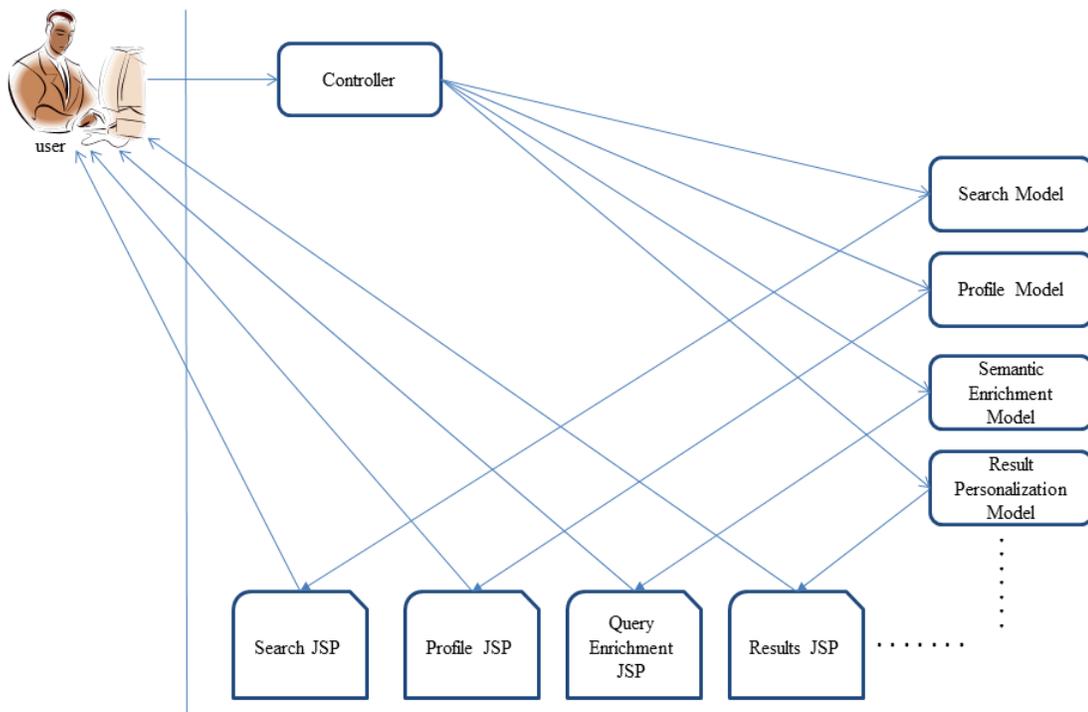


Figure 44 MVC Design of the System

For the query processing, “SearchAction” class receives the user’s query and dispatches the call to “UserContext” class to check if the user is authenticated. If so, it retrieves the user’s profile and transforms it into the shape of an instance of “CreateProfileForm” class.

⁹

https://en.wikibooks.org/wiki/Computer_Science_Design_Patterns/Model%E2%80%93view%E2%80%93controller

After loading the user’s profile, “SearchAction” dispatches the call to the “UserQueryProcessor,” which does the tasks required to process the query semantically. It then formulates an instance of the class “ProcessedUserQuery,” which contains the processed semantic information of the query, such as the concepts and relations that are found in the query. The query is then dispatched to “QuestionTemplateManager,” which matches the user’s query to the nearest query template. Figure 45 shows the class diagram for classes needed for query processing.

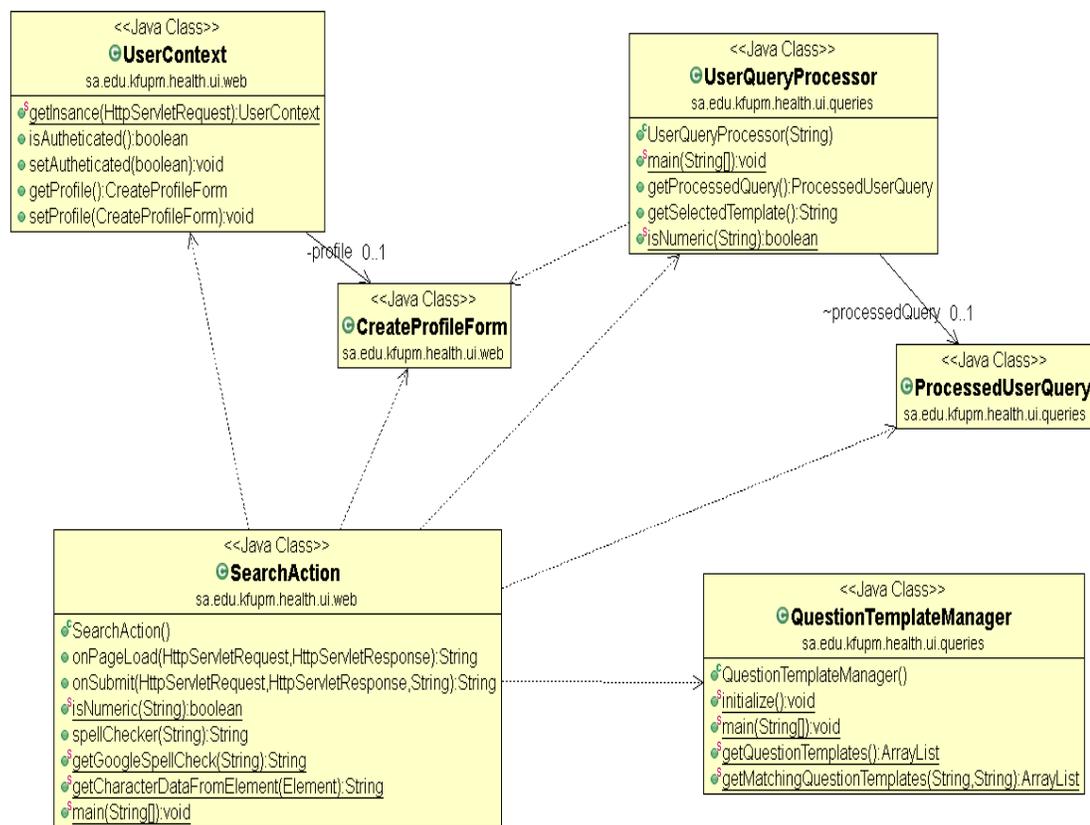


Figure 45 Query Processing Class Diagram

To reason the query, we deal with reasoning engine, which provides reasoning templates. Out of the reasoning templates, we create templates and widgets that correspond to a single semantic query. The templates are based on the question type, relations, and concepts founded in the user’s query. Each template contains a number of widgets.

“QBox” class is a notation for the reasoning template. We create the required number of templates based on the queries’ varieties. For example, “ListPositiveQBox” class represents the food list that has a positive relationship with a certain disease. This class corresponds to “WidgetListPositive” class, which is used to post-process and filter the results based on the user’s profile. Figure 46 presents the class diagrams for the required classes in the results personalization. It shows only a few examples of the widgets and templates, while we have many templates that represent different relations and question types.

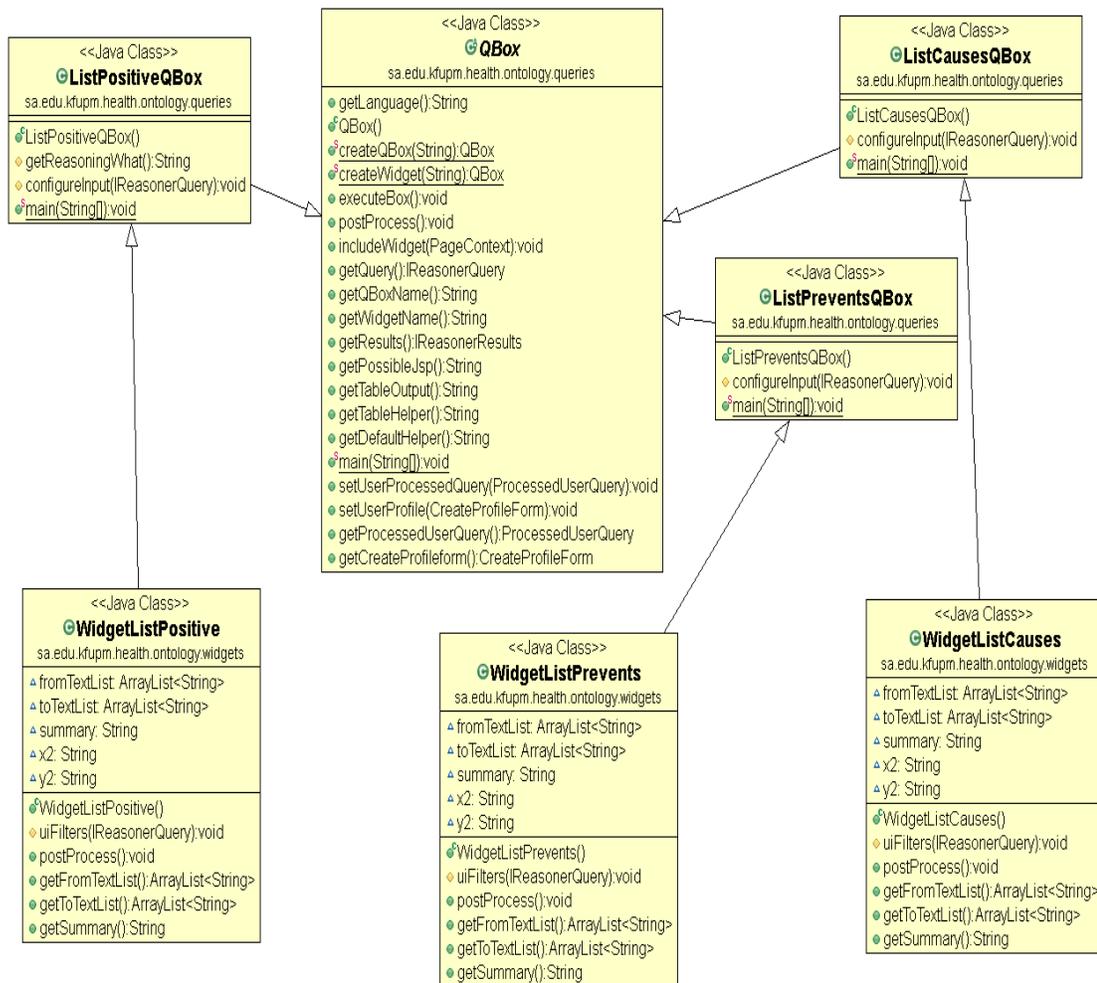


Figure 46 Results Personalization Class Diagram

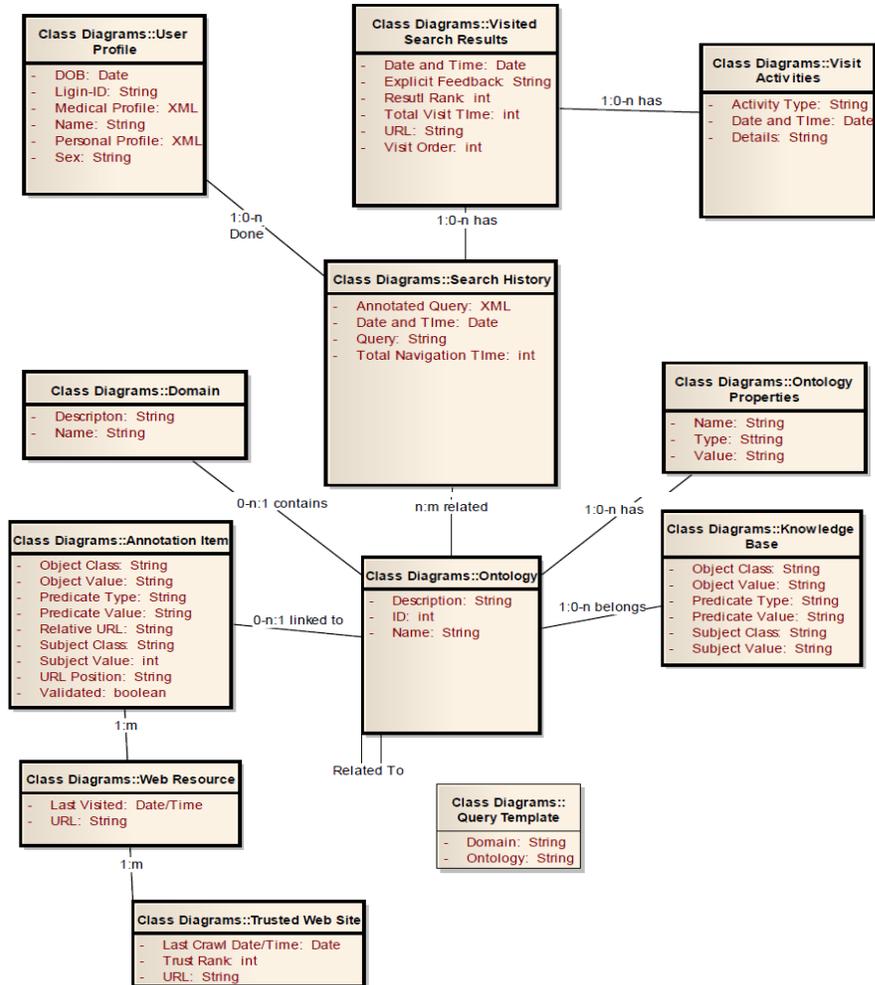


Figure 47 Class Diagram

Figure 47 shows the class diagram that consists of these different classes:

- User Profile: contains the information about the user.
- Visited Search Results: contains the information about the visited results.
- Visit Activities: logs the activities the user does when visiting the result.
- Search History: logs the user’s queries and annotated queries for future use.
- Domain: saves the domains of the ontology
- Ontology Properties: represents the ontology properties and their values.
- Ontology: describes the domain ontologies.

- Knowledgebase: saves the annotated information.
- Annotated item: represents a single annotated item.
- Web resources: documents the web resources used in the annotation.
- Trusted websites: ranks the web sources based on their trust level.
- Query template: saves the template used in query manipulation.

8.4 Tools and Programming Languages Used

Below are the technical specifications of the developed software.

8.4.1 Hardware and Software Interfaces

The hardware interfaces of the system are handled by the underlining operating system.

The system is developed using Java virtual machine, which is a machine-independent platform. The developed application utilizes the Java platform for hardware interface functionality that can work under deferent hardware such as a PC, handheld assistance, and mobile phones that support Java Virtual Machine. The software interfaces are described in the following:

- The software is designed to run on Java Virtual Machine with the JBoss¹⁰ Application Server.
- The software is designed to run on the Apache¹¹ Tomcat web server 6.0.18.
- The software accesses mySQL¹² database for the following features:
 - o adding and updating the user's query history;
 - o storing the visited results;

¹⁰ <http://www.jboss.org/>

¹¹ <http://tomcat.apache.org/>

¹² <http://www.mysql.com/>

- storing query templates; and
- maintaining user's action and activity logs.
- Ontology OWL and store tools are used to access the domain ontologies.

8.4.2 Programming Languages

The programming languages used to build the system include:

- Java Enterprise Edition (JAVA EE:): which is a part of the Java Platform for developing and running distributed multi-tier architecture Java applications, based largely on modular software components running on an application server.
- Hyper Text Markup Language (HTML) and Extensible Markup Language (XML): which are the predominant markup languages for webpages. They provide the means to describe the structure of text-based information in a document and to supplement that text with interactive forms, embedded images, and other objects.
- JavaScript: A client-side scripting language used to create dynamic web content and user interface.

8.4.3 Development Tools

The development tools used in the system include:

- Apache Tomcat 6.0.18 Server: Apache Tomcat is a Servlet container developed by the Apache Software Foundation (ASF). Tomcat implements the Java Servlet and the JavaServer Pages (JSP) specifications from Sun Microsystems and provides a "pure Java" HTTP web server environment for Java code to run in.
- ECLIPSE J2EE¹³: Eclipse is a toolkit designed for the creation of complex projects, providing fully dynamic web application utilizing Enterprise Java Beans

¹³ <https://www.eclipse.org>

(EJBs). This consists of EJB tools, CMP, data mapping tools, and a universal test client designed to aid testing of EJBs.

- Jena¹⁴: Jena is a Java framework for building Semantic Web applications. Jena provides a collection of tools and Java libraries to help you to develop the Semantic Web and linked-data apps, tools, and servers.
- Log4J¹⁵: Apache Log4j is a Java-based logging API. It allows the developer to control which log statements are output with arbitrary granularity. It is fully configurable at runtime using external configuration files.

8.4.4 Semantic Web Tools

The Semantic Web tools used in the software include:

- TopRaid Composer Ontology Editor¹⁶: which is used to represent the ontology. TopRaid Composer is an enterprise-class modeling environment for developing Semantic Web ontologies and building semantic applications. Fully compliant with W3C standards, Composer offers comprehensive support for developing, managing, and testing configurations of knowledge models and their instance knowledgebases. TopRaid Composer is the leading industrial-strength RDF editor and OWL ontology editor, as well as the best SPARQL tool on the market.
- OWLIM reasoning¹⁷: which is used to reason the semantic queries. OWLIM is the most scalable semantic repository. It includes triple store, an inference engine, and the SPARQL query engine. It is packaged as a storage and inference layer (SAIL) for the Sesame RDF database. OWLIM uses the TRREE engine to perform RDFS,

¹⁴ <http://jena.apache.org/>

¹⁵ <http://logging.apache.org/log4j/>

¹⁶ <http://www.topquadrant.com/tools/ide-topbraid-composer-maestro-edition/>

¹⁷ <https://confluence.ontotext.com/display/OWLIMv43/OWLIM-Lite+Reasoner>

OWL DLP, OWL Horst reasoning, and OWL 2 RL. The most expressive language supported is OWL 2 RL, containing RDFS. OWLIM offers configurable reasoning support and performance.

- Sesame RDF Store: it is used to store the semantic annotated data. Sesame is an open-source framework for querying and analyzing RDF data. Sesame's API differs from comparable solutions in that it offers a stackable interface through which functionality can be added and the storage engine is abstracted from the query interface [1].

8.5 Implementation Details

A web-based system has been developed to implement the proposed framework. The semantic techniques are used for reasoning and semantic storage, such as OWLIM¹⁸ and Sesame RDF.¹⁹ Semantic techniques are integrated with Java J2EE, HTML, and JavaScript technologies to show the user-friendly front end. The multi-agent framework JADE²⁰ is used to communicate between the agents in addition to AgentOWL,²¹ which is an agent library that supports the RDF/OWL model based on the Jena framework. A Java spell-checking library is used in addition to WordNet²² for synonyms.

8.5.1 Snapshots

Figure 48, Figure 49, and Figure 50 show the snapshot screens of the implemented system.

¹⁸ <https://www.ontotext.com/owlim>

¹⁹ <http://www.openrdf.org/>

²⁰ <http://jade.tilab.com/>

²¹ <http://ups.savba.sk/~miso/AgentOWL/doc/>

²² <http://wordnet.princeton.edu>

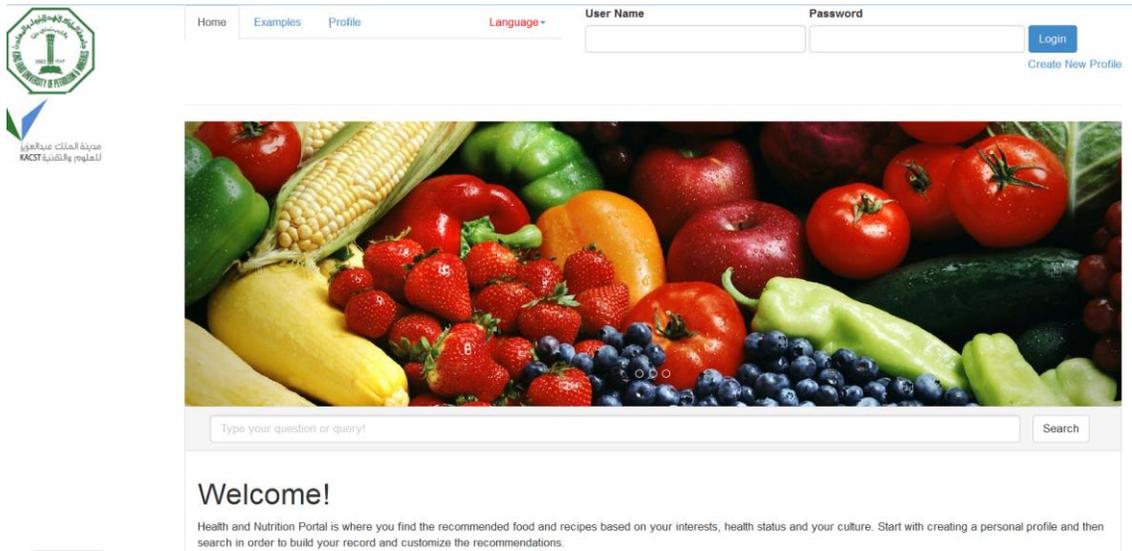


Figure 48 Portal Main Screen Snapshot

Ginger, coffee, garlic, apples, mushrooms, pea and chamomile tea can **prevent** diabetes. Below are more details with snippet from the Web reference.

Ginger root prevents gestational diabetes (3 records)

Source Text:
 Research has found that **ginger** may be beneficial in the **fight against diabetes**.

Source Link:
<http://www.healthdiaries.com/eatthis/ginger>

Source Text:
 Research has found that **ginger** may be beneficial in the **fight against diabetes**.

Source Link:
<http://www.healthdiaries.com/eatthis/blood-sugar>

Source Text:
 Research has found that **ginger** may be beneficial in the **fight against diabetes**.

Source Link:
<http://www.healthdiaries.com/eatthis/archives.html>

Barley prevents gestational diabetes

Source Text:
 I have read above blog of 8 health benefit of **barley** grass powder. It contains as 70 different different minerals, numerous vitamins, enzymes and antioxidants and protein. It **prevent** cancer **diabetes** and balance cholesterol and more benefit provide in our body. So I suggested every one take this one in our increase long lifestyle.

Source Link:
<http://www.healthdiaries.com/eatthis/8-health-benefits-of-barley-grass.html>

Your question analysis

What **food** can prevent **diabetes**?

Found Concept	Value
Question: Subjective	.. What ...
Relation: prevents	.. can prevent ..
Food	.. food ..
Disease	.. diabetes ..

Your personal profile

Property	Value
Location, culture and religion	Saudi Arabia, Saudi culture and Islam
Diseases	Diabetes, lack of anzimes.

[More information in your profile](#)

Figure 49 Example of the Results Page

Please fill as much as you can in order to refine your query better. Note that these information are secured and not to be shared with 3rd party.

Logon information

Basic User Information

Male
 Female

Yes No
 Yes No
 Yes No

Food Preferences

Basic Health Information

Calculated BMI

Activity Level

Check the activity level that best matches your lifestyle.

Sedentary

- At work - you work in an office
- undefined
- Exercise - you don't exercise regularly

Light Activity

- At work - you walk a lot
- undefined
- Exercise - you participate in light exercise or take long walks

Moderate Activity

- At work - you are very active much of the day
- undefined
- Exercise - you exercise several times a week and push yourself pretty hard

Very Active

- At work - you hold a labor-intensive job such as construction worker or bicycle messenger
- undefined
- Exercise - you participate in physical sports such as jogging or mountain-biking each day

Medical Profile

Health Goals

Figure 50 User profile screen snapshot

8.5.2 Agents Implementation

We use JADE²³ to implement the backend communications between agents to facilitate communication and benefit from the agent-based modeling. JADE is considered middleware providing a platform and an API for developing agent-based systems. Once the agent is created, it is registered in the JADE Directory Facilitator (DF) to communicate with the rest of the agents. The JADE DF facilitates finding agents and provides an idea about the services provided by the agent, which can help another agent in achieving its goal. This is called Yellow Pages service in JADE DF. An agent may not have previous knowledge about the other agents. Figure 51 gives an idea of how JADE is a middle layer to get agents to talk with each other.

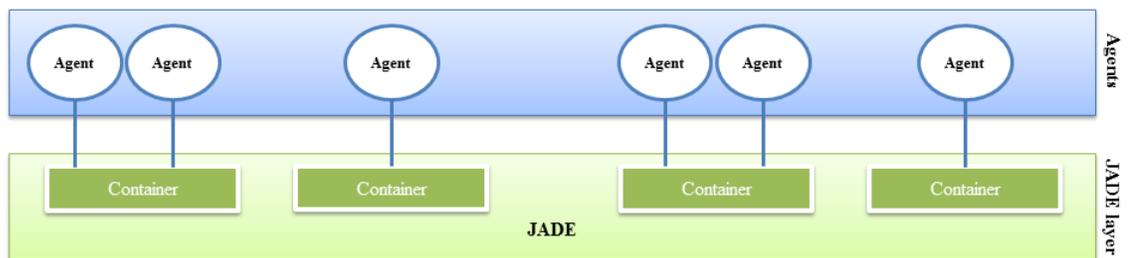


Figure 51 JADE Layer

The following JADE packages and classes were used:

- *Jade.core* package: which contains *Agent* class, the main class in the framework. This class is extendable to the main class of each agent.
- *Behavior* class: which is under the *jade.core.behaviours* package. This class takes care of the agent's tasks and determines the behavior and functions of the agent. The main advantage of this class is the concurrent execution of multiple instances.

²³ <http://jade.tilab.com/>

- *jade.lang.acl* package: which is involved in the communication between agents using *Agent Communication Language*, which is implemented with reference to the Foundation for Intelligent Physical Agents (FIPA) standard specifications.²⁴
- *jade.domain* package: which is involved in agent management activities defined in the FIPA standard and specifically an agent management system (AMS) agent that controls the agent platform, and directory facilitator (DF) agents provide directory for the agent services.

The AMS agent plays a major role in controlling the access and the use of agents. There is only one instance of AMS in a single-agent platform. The AMS provides other services such as life-cycle service and managing a directory that contains agent identifiers (AID) and agent state. Any agent should enroll itself with AMS, which provides the agent with an ID called AID. Agent communication channel (ACC), also called the message transport system, controls all communications between the agents within a platform and from external platforms. Figure 52 shows FIPA architecture.

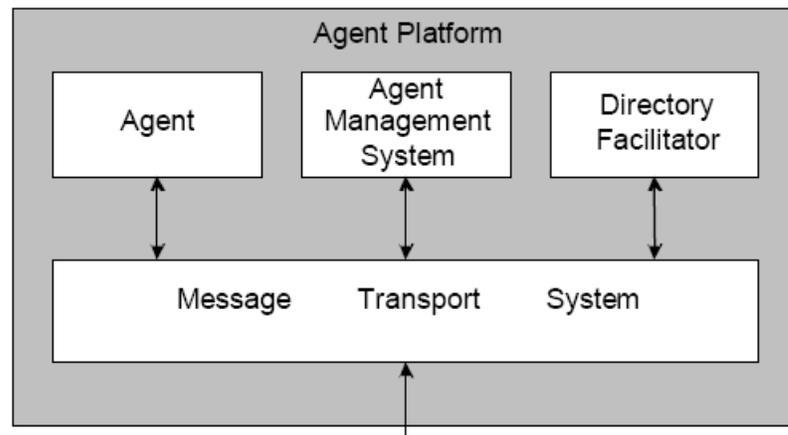


Figure 52 FIPA Specification, Source (141) Figure 2

²⁴ <http://www.fipa.org/>

We have used different agent operations through defining different agents for various behaviors:

- *RequestAgent*: which extends *Behavior* class involved in sending one request only. An example is when the interface agent sends the user's query to the semantic query manipulation agent. In this case, a new instance of this class is created when the user's interface agent receives new query.
- *ReplyAgent*: which extends *CyclicBehavior* class, a child of *Behavior* class with the advantage that it is alive while the program is alive, and this helps in listening to the events. The normal *Behavior* object is limited and executed one time only when requested. *ReplyAgent* provides services for other agents.

8.5.3 Query and Result Templates Implementation

Query templates are used to match the user's query after the semantic query manipulation process. The query templates define the input, the query, and the expected output, the results. We define these templates based on the analysis of the domain ontologies and knowledgebase.

We have used several ontologies, including health condition, which has three childhood diseases, body part, and body function. Then we have food, nutrition, recipe, user's profile, and culture. User's profile and culture are used to enrich the query. We also have support ontologies for serving size, which is related to food items, and daily needs, which is related to nutrition.

We also define a set of relations between different ontologies. TABLE 24, TABLE 25, TABLE 26, TABLE 27, TABLE 28, TABLE 29, and TABLE 30 show the defined relations and some examples of English and Arabic terms that are used for queries. We

use the lookup dictionary to match with user's input. We have seven relations defined: has positive effect, has negative effect, causes, prevents, treats, contains, and details.

TABLE 24 Definition of the Relation: HAS_POSTIVE_EFFECT

Relation name	HAS_POSTIVE_EFFECT
Relation between	(1) Food-Health Condition, (2) Nutrient-Health Condition, (3) Recipe-Health Condition
Examples of English terms	improve, strength, suit
Examples of Arabic terms	يناسب, يكفي, يفيد

TABLE 25 Definition of the Relation: HAS_NEGATIVE_EFFECT

Relation name	HAS_NEGATIVE_EFFECT
Relation between	(1) Food-Health Condition, (2) Nutrient-Health Condition, (3) Recipe-Health Condition
Examples of English terms	worsen, destroy, go bad
Examples of Arabic terms	يسبب, يضر, يفسد

TABLE 26 Definition of the Relation: CAUSES

Relation name	CAUSES
Relation between	(1) Food-Disease, (2) Nutrient-Disease, (3) Recipe-Disease
Examples of English terms	cause, lead to, result
Examples of Arabic terms	يسبب, يؤدي, ينتج

TABLE 27 Definition of the Relation: PREVENTS

Relation name	PREVENTS
Relation between	(1) Food-Disease, (2) Nutrient-Disease, (3) Recipe-Disease
Examples of English terms	prevent, stop, block
Examples of Arabic terms	يوقف, يستبعد, يتجنب

TABLE 28 Definition of the Relation: TREATS

Relation name	TREATS
Relation between	(1) Food-Disease, (2) Nutrient-Disease, (3) Recipe-Disease
Examples of English terms	treat, attend to, nurse
Examples of Arabic terms	يعالج, يشفي, يداوي

TABLE 29 Definition of the relation: CONTAINS

Relation name	CONTAINS
Relation between	(1) Food-Recipe, (2) Nutrient- Recipe, (3) Nutrient-Food
Examples of English terms	contain, include, have
Examples of Arabic terms	يحتوي, يتكون من, يشتمل

TABLE 30 Definition of the Relation: DETAILS

Relation name	DETAILS
Relation between	(1) Recipe-Food, (2) Recipe-Nutrient, (3) Food-Nutrient
Examples of English terms	ingredient, components
Examples of Arabic terms	مكونات, محتويات

Questions are classified into three categories: LIST-questions, IS-questions, and QUANTITY-questions. The question type is important as it leads to the correct way to answer the query. TABLE 31, TABLE 32, and TABLE 33 show the defined questions with some trigger words for English and Arabic. To identify the question type, we take the first two terms in the query and compare them with the defined list of terms for each question type.

TABLE 31 Definition of LIST-Questions

Question type	LIST-question
Examples of English terms	what, list, which
Examples of Arabic terms	ماذا, عدد, أي, ما
Expected output	Lists items that match the question criteria

TABLE 32 Definition of IS-Questions

Question type	IS-question
Examples of English terms	do, is, are, can
Examples of Arabic terms	أليس, هل
Expected output	Confirms by yes or no based on the question criteria

TABLE 33 Definition of QUANTITY-Questions

Question type	QUANTITY-question
Examples of English terms	how, how much
Examples of Arabic terms	بكم, كم
Expected output	Returns the quantity based on the question criteria

TABLE 34 Examples of Query and Result Templates

Template name	Input	Output
LIST_FOOD_PREVENTS_DISEASE	Question type (LIST), Relation (prevents), Disease Food (optional)	List (Food, prevents, Disease)
LIST_FOOD_CAUSES_DISEASE	Question type (LIST), Relation (causes), Disease, (optional)	List (Food, causes, Disease) Food
IS_FOOD_TREATS_DISEASE	Question type (IS), Relation (treats), Disease, Food	Yes/No (Food, treats, Disease)
LIST_FOOD_DETAILS	Question type (LIST), Relation (details), Food	List (Nutrition, details, Food)
IS_FOOD_CONTAINS_NUTRITION	Question type (IS), Relation (contains), Food, Nutrition	Yes/No (Food, contains, nutrition)

After we defined the domain ontologies, the possible relations between concepts and instance, and finally the categories of the question types, TABLE 34 shows some examples of the query and result templates. These templates are used to match the user's query and then return the results.

CHAPTER 9

EXPERIMENT AND ANALYSIS

In this chapter, we show first a complete example of the whole system and the experimental results of different experiments.

9.1 Complete Example

The objective of this experiment is to run a complete example to test the following:

- The semantic query understanding in detail
- The transformation of the semantic information
- The results retrieval (non-personalized)
- The user's profile
- The query enrichment
- The personalized retrieval (filtering and ranking)

We show a complete example where the user enters a query, and then the query is semantically manipulated. Then we show how the system retrieves the results in two scenarios, one with a user's profile and the other without a user's profile. The query is entered in English, "What food can help in preventing diabetes?" and in Arabic, "ما هي الأطعمة التي تساعد على تجنب السكري؟"

9.1.1 Query Manipulation Example

The semantic query manipulation process has a number of steps, as shown in TABLE 35. For part-of-speech tagging, we use English and Arabic taggers based on the Stanford Log-linear Part-Of-Speech Tagger,²⁵ which is based on the notations of the Penn Treebank P.O.S. Tags²⁶ in addition to some notations used for Arabic language, such as DTNN, which means the noun starts with “ال” “al”.²⁷ TABLE 36 shows the descriptions of the used tags.

TABLE 35 Example of Semantic Query Manipulation

Input	What food can help in preventing	ما هي الأطعمة التي تساعد على تجنب السكري؟
Query	diabetes?	
Step 1: Language detection		
Input	What food can help in preventing	ما هي الأطعمة التي تساعد على تجنب السكري؟
	diabetes?	
Output	(LANGUAGE: English)	(LANGUAGE: Arabic)
Step 2: Terms tokenizing		
Input	What food can help in preventing	ما هي الأطعمة التي تساعد على تجنب السكري؟
	diabetes?	
Output	What, food, can, help, in, preventing, diabetes, ?	تجنب, على, تساعد, التي, الأطعمة, هي, ما, السكري

²⁵ <http://nlp.stanford.edu/software/tagger.shtml>

²⁶ https://www.ling.upenn.edu/courses/Fall_2003/ling001/penn_treebank_pos.html

²⁷ <http://nlp.stanford.edu/software/parser-arabic-faq.shtml>

Step 3: Spell checking		
Input	What, food, can, help, in, preventing, diabetes, ?	تجنب, على, تساعد, التي, الأطعمة, هي, ما, السكري, ؟
Output	Null (i.e., all words are spelled correctly)	
Step 4: Part of speech (POS) tagging		
Input	What, food, can, help, in, preventing, diabetes, ?	تجنب, على, تساعد, التي, الأطعمة, هي, ما, السكري
Output	What/WDT food/NN can/MD help/VB in/IN preventing/VBG diabetes/NN	ما/WP هي/PRP الأطعمة/DTNN التي/WP تساعد/VBP على/IN تجنب/NN السكري/DTNN
Step 5: Question type classification		
Input	What, food	هي, ما
Output	List question (“what” belongs to pre-defined list of list question type)	
Step 6: Measurement identification		
Input	What, food, can, help, in, preventing, diabetes, ?	تجنب, على, تساعد, التي, الأطعمة, هي, ما, السكري
Output	Null (i.e., no measurement quantity, no measurement unit)	
Step 7: Noise words removal		
Input	What, food, can, help, in, preventing, diabetes, ?	تجنب, على, تساعد, التي, الأطعمة, هي, ما, السكري
Output	food, help, preventing, diabetes	السكري, تجنب, تساعد, الأطعمة
Step 8: Semantic named entity recognition		
Input	food, help, preventing, diabetes	السكري, تجنب, تساعد, الأطعمة

Output	Food (concept: FOOD), diabetes (instance: DISEASE: DIABETES)	الأطعمة (concept: FOOD), السكري (instance: DISEASE: DIABETES)
--------	---	--

Step 9: Morphological analysis

Input	<VB>help, <VBG>preventing	<VB>تجنب, <VB>تساعد
Output	help, prevent	جنب, ساعد

Step 10: Relation identification

Input	help, (preventing, prevent)	(جنب, تجنب), (ساعد, ساعد)
Output	help (relation: POSITIVE), preventing (relation: PREVENT)	تجنب (relation: POSITIVE), (relation: PREVENT)

Step 11: Defined terms identification

Input	Null (no remaining words)	
Output	Null (no other term identified)	

Step 12: Non-identified terms processing (using context, patterns, synonymous)

Input	Null (no remaining words)	
Output	Null (no post processing terms found)	

Step 13: Ambiguity resolution (using context, patterns, weighted ENR)

Input	What food can help in preventing diabetes?, Food (concept: FOOD), diabetes (instance: DISEASE: DIABETES), help (relation: POSITIVE), preventing (relation: PREVENT)	, ما هي الأطعمة التي تساعد على تجنب السكري؟ السكري (concept: FOOD), (instance: DISEASE: DIABETES), تجنب (relation: POSITIVE), (relation: PREVENT)
Output	Food (concept: FOOD), diabetes	السكري (concept: FOOD), الأطعمة

	(instance: DISEASE: DIABETES), preventing (relation: PREVENT)	(instance: DISEASE: DIABETES), تجنب (relation: PREVENT)
Step 14: Query template matching		
Input	(concept: FOOD), (instance: DISEASE: DIABETES), (relation: PREVENT)	
Output	(template: TEMPLATE_LIST_FOOD_PREVENTS_DISEASE) (instance: DISEASE: DIABETES)	
Semantic	(LANGUAGE: English)	(LANGUAGE: Arabic)
query	(template: LIST_FOOD_PREVENTS_DISEASE), (instance: DISEASE:	
output	DIABETES)	

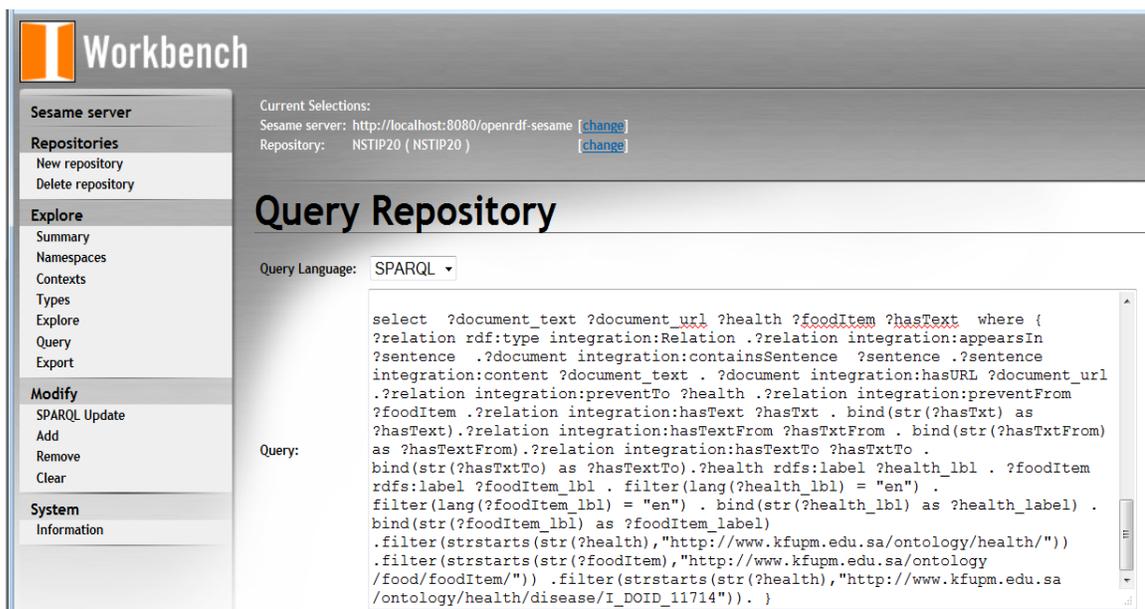
TABLE 36 Part-of-Speech Tags with Their Descriptions

Tag	Description	English example	Arabic example
WDT	Wh-determiner	What	
NN	Noun, singular or mass	Food, diabetes	تجنب
MD	Modal	Can	
VB	Verb, base form	Help	
IN	Preposition or subordinating conjunction	In	على
VBG	Verb, gerund or present participle	Preventing	
WP	Wh-pronoun		التي, ما
PRP	Personal pronoun		هي
VBP	Verb, non-3rd person singular present		تساعد
DTNN	Noun, starts with "al" for Arabic terms		السكري, الأظعمة

In the next section, we show the detailed steps of retrieving the results in two options: with and without a user's profile.

9.1.2 Results Retrieval without User's Profile Example

The user can use the system without a profile, but this scenario will show that the user will miss many features such as query enrichment and personalized retrieval. The search results come in a semantic representation, as we show the SPARQL query executed in Figure 53. The semantic results are shown in Figure 54.



The screenshot displays the 'Workbench' interface. On the left is a navigation sidebar with sections: 'Sesame server', 'Repositories' (with sub-items 'New repository' and 'Delete repository'), 'Explore' (with sub-items 'Summary', 'Namespaces', 'Contexts', 'Types', 'Explore', 'Query', 'Export'), 'Modify' (with sub-items 'SPARQL Update', 'Add', 'Remove', 'Clear'), and 'System' (with sub-item 'Information'). The main area is titled 'Query Repository' and shows 'Current Selections' for 'Sesame server' (http://localhost:8080/openrdf-sesame) and 'Repository' (NSTIP20). Below this, the 'Query Language' is set to 'SPARQL'. A text area contains the following SPARQL query:

```
select ?document_text ?document_url ?health ?foodItem ?hasText where {
?relation rdf:type integration:Relation .?relation integration:appearsIn
?sentence .?document integration:containsSentence ?sentence .?sentence
integration:content ?document_text . ?document integration:hasURL ?document_url
.?relation integration:preventTo ?health .?relation integration:preventFrom
?foodItem .?relation integration:hasText ?hasText . bind(str(?hasText) as
?hasText).?relation integration:hasTextFrom ?hasTxtFrom . bind(str(?hasTxtFrom)
as ?hasTextFrom).?relation integration:hasTextTo ?hasTxtTo .
bind(str(?hasTxtTo) as ?hasTextTo).?health rdfs:label ?health_lbl . ?foodItem
rdfs:label ?foodItem_lbl . filter(lang(?health_lbl) = "en") .
filter(lang(?foodItem_lbl) = "en") . bind(str(?health_lbl) as ?health_label) .
bind(str(?foodItem_lbl) as ?foodItem_label)
.filter(strstarts(str(?health), "http://www.kfupm.edu.sa/ontology/health/"))
.filter(strstarts(str(?foodItem), "http://www.kfupm.edu.sa/ontology
/food/foodItem/")) . filter(strstarts(str(?health), "http://www.kfupm.edu.sa
/ontology/health/disease/I_DOID_11714")). }
```

Figure 53 SPARQL Semantic Query

The screenshot shows the OpenRDF Workbench interface. The top navigation bar includes 'Workbench' and 'OpenRDF'. The left sidebar contains navigation options like 'Sesame server', 'Repositories', 'Explore', 'Modify', and 'System'. The main content area displays a 'Query Result (7)' with a table of 7 rows and 5 columns. The columns are 'Document_text', 'Document_url', 'Health', 'FoodItem', and 'HasText'. The table contains various search results related to diabetes and health benefits.

Document_text	Document_url	Health	FoodItem	HasText
"Research has found that ginger may be beneficial in the fight against diabetes."@en	"http://www.healthdiaries.com/earthis/ginger"^^xsd:string	<http://www.kfupm.edu.sa/ontology/health/disease/I_DOID_11714>	<http://www.kfupm.edu.sa/ontology/food/foodItem/I11216>	"fight against"
"Research has found that ginger may be beneficial in the fight against diabetes."@en	"http://www.healthdiaries.com/earthis/blood-sugar"^^xsd:string	<http://www.kfupm.edu.sa/ontology/health/disease/I_DOID_11714>	<http://www.kfupm.edu.sa/ontology/food/foodItem/I11216>	"fight against"
"Research has found that ginger may be beneficial in the fight against diabetes."@en	"http://www.healthdiaries.com/earthis/archives.html"^^xsd:string	<http://www.kfupm.edu.sa/ontology/health/disease/I_DOID_11714>	<http://www.kfupm.edu.sa/ontology/food/foodItem/I11216>	"fight against"
"Studies have found that evening primrose oil may also help protect from diabetic neuropathy, a nerve disorder often affecting people with diabetes that causes tingling, pain, numbness, and other symptoms in the legs and feet."@en	"http://www.healthdiaries.com/earthis/8-health-benefits-of-evening-primrose-oil.html"^^xsd:string	<http://www.kfupm.edu.sa/ontology/health/disease/I_DOID_11714>	<http://www.kfupm.edu.sa/ontology/food/foodItem/I04034>	"may also help protect"
"I have read above blog of 8 health benefit of barley grass powder. It contains as 70 different minerals, numerous vitamins, enzymes and antioxidants and protein. It prevent cancer, diabetes and balance cholesterol and more benefit provide in our body. so i suggested every one take this one in our increase long lifestyle."@en	"http://www.healthdiaries.com/earthis/8-health-benefits-of-barley-grass.html"^^xsd:string	<http://www.kfupm.edu.sa/ontology/health/disease/I_DOID_11714>	<http://www.kfupm.edu.sa/ontology/food/foodItem/I20004>	"prevent"
"A review of 18 studies, involving 450,000 people, published in Archives of Internal Medicine, found that each additional cup of coffee consumed per day lowered the risk of diabetes by 7%."@en	"http://www.healthdiaries.com/earthis/6-health-benefits-of-decaf-coffee.html"^^xsd:string	<http://www.kfupm.edu.sa/ontology/health/disease/I_DOID_11714>	<http://www.kfupm.edu.sa/ontology/food/foodItem/I14201>	"lowered the risk of"
"Garlic has been shown to protect rats from diabetes complications such as retinopathy, kidney disease, cardiovascular disease, and neuropathy."@en	"http://www.healthdiaries.com/earthis/20-health-benefits-of-garlic.html"^^xsd:string	<http://www.kfupm.edu.sa/ontology/health/disease/I_DOID_11714>	<http://www.kfupm.edu.sa/ontology/food/foodItem/I11215>	"to protect"

Figure 54 Semantic Results

We render the semantic results and show them to the user in a user-friendly way even if the user has no profile. Figure 55 shows the results based on the limited dataset we have. The retrieved results have seven records, three for the same fact, which are collected together, and three single ones. We show on the left the facets for each food group and food item. Facets are used to help the user to explore or filter the results based on the user's needs. Users are motivated to create a profile to benefit from the personalization techniques we provide. If the users have a profile they can mark their preferred food and the system automatically learns their preferences through explicit and implicit feedback.

What food can help in preventing diabetes? Search

Food Groups

- Vegetables
- Drinks
- Grain

Food Items

- Ginger
- barley
- coffee
- Garlic

6 Records Found

Ginger, barley, coffee and barley help in preventing diabetes. Details are below.

Ginger prevents diabetes (3 records)

Source Text:
 Research has found that **ginger** may be beneficial in the **fight against diabetes**.
[Source Link-1](#) - [Source Link-2](#) - [Source Link-3](#)

Barley prevents diabetes

Source Text:
 I have read above blog of 8 health benefit of **barley** grass powder. It contains as 70 different different minerals, numerous vitamins, enzymes and antioxidants and protein. it **prevent** cancer, **diabetes** and balance colestrol and more benefit provide in our body. so i suggested every one take this one in our increase long lifestyle.
[Source Link](#)

Coffee prevents diabetes

Source Text:
 A review of 18 studies, involving 450,000 people, published in Archives of Internal Medicine, found that each additional cup of **coffee** consumed per day **lowered the risk of diabetes** by 7%.
[Source Link](#)

Garlic prevents diabetes

Source Text:
Garlic has been shown **to protect** rats from **diabetes** complications such as retinopathy, kidney disease, cardiovascular disease, and neuropathy.
[Source Link](#)

Your question analysis

What food help in preventing diabetes?

Found Concept	Value
Question: List	.. What ...
Relation: prevents	.. preventing ..
Food	.. food ..
Disease	.. diabetes ..

Your personal profile

[Create your own profile](#)

Figure 55 Results without Personalization

9.1.3 Results Retrieval with User's Profile Example (Personalized Retrieval)

The user can use the system with a profile to show the advantage of using the profile in many things, such as query enrichment and personalized retrieval. For a simple experiment, for example, the user likes coffee and does not like grain. Coffee will take first place in the recommendation, while grain is pushed to the end. Figure 56 shows the personalized result the user retrieves and the ability to enhance the profile.

What food can help in preventing diabetes? Search

Food Groups

Vegetables

Drinks

Food Items

Ginger

coffee

Garlic

6 Records Found

Coffee, ginger, barley and barley help in preventing diabetes. Details are below.

Coffee prevents diabetes (Liked food)

Source Text:
A review of 18 studies, involving 450,000 people, published in Archives of Internal Medicine, found that each additional cup of **coffee** consumed per day **lowered the risk of diabetes** by 7%.

[Source Link](#)

Ginger prevents diabetes (3 records)

Source Text:
Research has found that **ginger** may be beneficial in the **fight against diabetes**.

[Source Link-1](#) - [Source Link-2](#) - [Source Link-3](#)

Garlic prevents diabetes

Source Text:
Garlic has been shown **to protect** rats from **diabetes** complications such as retinopathy, kidney disease, cardiovascular disease, and neuropathy.

[Source Link](#)

Barley prevents diabetes (Disliked food)

Source Text:
I have read above blog of 8 health benefit of **barley** grass powder. It contains as 70 different different minerals, numerous vitamins, enzymes and antioxidants and protein. It **prevent** cancer, **diabetes** and balance cholesterol and more benefit provide in our body. so i suggested every one take this one in our increase long lifestyle.

[Source Link](#)

Your question analysis

What food help in preventing diabetes?

Found Concept	Value
Question: List	.. What ...
Relation: prevents	.. preventing ..
Food	.. food ..
Disease	.. diabetes ..

Your personal profile

Property	Value
Disliked Food	Grain
Liked Food	Coffee
Diseases	Diabetes

[More information in your profile](#)

Figure 56 Results with Profile

Moreover, if users use an Arabic query they will get similar results in their own language, as shown in Figure 57.

ما هي الأطعمة التي تساعد على تجنب السكري؟

بحث

تحليل سؤالك

ما هي الأطعمة التي تساعد على تجنب السكري؟

المصطلح: القهوة

القيمة: سؤال تعداد: نوع السؤال .. ما هي ...

العلاقة: تجنب .. تجنب ..

الطعام: .. الأطعمة ..

المرض: .. السكري ..

وجدنا 6 نتائج

القهوة والزنجبيل والثوم تساعد على تجنب السكري. التفاصيل بالأسفل.

القهوة تساعد على تجنب السكري (مفضلة)

نص المصدر:

A review of 18 studies, involving 450,000 people, published in Archives of Internal Medicine , found that each additional cup of **coffee** consumed per day **lowered the risk of diabetes** by 7%

الزنجبيل تساعد على تجنب السكري (3 مصادر)

نص المصدر:

.Research has found that **ginger** may be beneficial in the **fight against diabetes**

الثوم تساعد على تجنب السكري

نص المصدر:

Garlic has been shown **to protect** rats from **diabetes** complications such as .retinopathy, kidney disease, cardiovascular disease, and neuropathy

الشعير تساعد على تجنب السكري (غير مرغوب)

نص المصدر:

I have read above blog of 8 health benefit of **barley** grass powder.It contains as 70 different different minerals, numerous vitamins, enzymes and antioxidants and protein.it **prevent** cancer, **diabetes** and balance colestrol and more benefit provide in .our body.so i suggested every one take this one in our increase long lifestyle

مجموعات الطعام

الحبوب

المشروبات

الطعام

الزنجبيل

القهوة

الثوم

ملفك الشخصي

الخاصية	القيمة
الطعام غير المفضل	الحبوب
الطعام المفضل	القهوة
الأمراض	السكري

معلومات أكثر

Figure 57 Arabic Personalized Results

9.2 String-Matching Experiment

We tested the accuracy of 23 string-matching algorithms. We took one term from the knowledgebase, an instance of a concept, and tried different variations of the term and other unrelated terms. We have tested both English and Arabic terms to select the appropriate string-matching algorithm that fits both language.

In the experiment, we have one input term from the user’s query and one indexed term from the domain ontologies knowledgebase. We observe how each string-matching algorithm correlates both the input term and the indexed term. We do this by using a normalized version of Levenshtein distance metric,²⁸ which measures the distance between the input term and the indexed term. For example, the Levenshtein distance between “fitting” and “getting” is 2 since there are two characters not matching. The Levenshtein distance metric is represented mathematically using the following formula²⁹:

$$\text{lev}_{a,b}(i, j) = \begin{cases} \max(i, j) & \text{if } \min(i, j) = 0, \\ \min \begin{cases} \text{lev}_{a,b}(i - 1, j) + 1 \\ \text{lev}_{a,b}(i, j - 1) + 1 \\ \text{lev}_{a,b}(i - 1, j - 1) + 1_{(a_i \neq b_j)} \end{cases} & \text{otherwise.} \end{cases}$$

Where $1_{(a_i \neq b_j)}$ is an indicator function that equals 0 when $a_i = b_j$ and equals 1 otherwise. The lowest value of the Levenshtein distance is the difference between the two strings’ sizes, while the highest value is the length of the string with more characters. Hence, the distance equals 0 if the two strings are exactly equal.

We use the normalized form of Levenshtein distance where we divide the calculated value per the above equation by the number of characters of the indexed term to make the lower limit is 0 and the upper limit 1. This helps us when comparing different algorithms with various words that have different sizes. Below is the equation we use for calculating the normalized Levenshtein distance:

$$\text{lev}_{a,b}^{\text{normalized}}(i, j) = \text{complement} \frac{\text{lev}_{a,b}(i, j)}{\text{length}(i)}$$

²⁸ <http://www.levenshtein.net/>

²⁹ http://en.wikipedia.org/wiki/Levenshtein_distance

where i is the indexed term and j is the input term. The complement is with 1. If the new value is 0, that means it is not matched while it is 1; that means it matches.

The implemented algorithms³⁰ are compared in TABLE 37 for the indexed term “التفاحة”, which is the Arabic label of the “apple” in the domain ontologies and knowledgebase.

TABLE 37 String-Matching Algorithms Performance

Algorithm/input term	تفاحة	تفاحتي	برتقالة	Result
BlockDistance	0.000	0.000	0.000	N
ChapmanLengthDeviation	0.714	0.857	1.000	N
ChapmanMatchingSoundex	0.000	0.000	0.000	N
ChapmanMeanLength	0.093	0.100	0.107	N
ChapmanOrderedNameCompoundSimilarity	0.500	0.333	0.143	<u>Y</u>
CosineSimilarity	0.000	0.000	0.000	N
DiceSimilarity	0.000	0.000	0.000	N
EuclideanDistance	0.000	0.000	0.000	N
JaccardSimilarity	0.000	0.000	0.000	N
Jaro	0.707	0.663	0.000	<u>Y</u>
JaroWinkler	0.707	0.663	0.000	<u>Y</u>
Levenshtein	0.714	0.429	0.429	YN
MatchingCoefficient	0.000	0.000	0.000	N
MongeElkan	1.000	0.667	0.286	<u>Y</u>
NeedlemanWunch	0.857	0.643	0.714	YN

³⁰ <http://www.coli.uni-saarland.de/courses/LT1/2011/slides/stringmetrics.pdf>

OverlapCoefficient	0.000	0.000	0.000	N
QGramsDistance	0.625	0.235	0.111	<u>Y</u>
SmithWatermanGotoh	1.000	0.667	0.286	<u>Y</u>
SmithWatermanGotohWindowedAffine	1.000	0.667	0.286	<u>Y</u>
SmithWaterman	1.000	0.667	0.286	<u>Y</u>
Soundex	0.000	0.000	0.000	N
[TagLinkToken_Tr_0.3]	0.343	0.248	0.429	N
LetterPairSimilarity	0.800	0.545	0.167	<u>Y</u>

We ran the experiment on Arabic text as it indicates how powerful the algorithm is. We observe that few of the algorithms did a good job for Arabic text. These are marked with “Y” where we chose the best one to use in our query manipulation. In addition to Levenshtein distance, other measures and tools that give the results of these measures include Java libraries. Examples include Simmetrics,³¹ which is an open-source Java-based library for similarity metric techniques, and Second String,³² which is another open-source Java-based library for approximate string-matching techniques. Figure 59, Figure 60, and Figure 61 show the experimental examples run using Simmetrics and Second String. In these two implementations, 1 means it is matching, while 0 means it is not matching. Figure 58 shows all the metrics equal to 1 when we pass two exact strings.

³¹ <http://sourceforge.net/projects/simmetrics/>

³² <http://secondstring.sourceforge.net/>

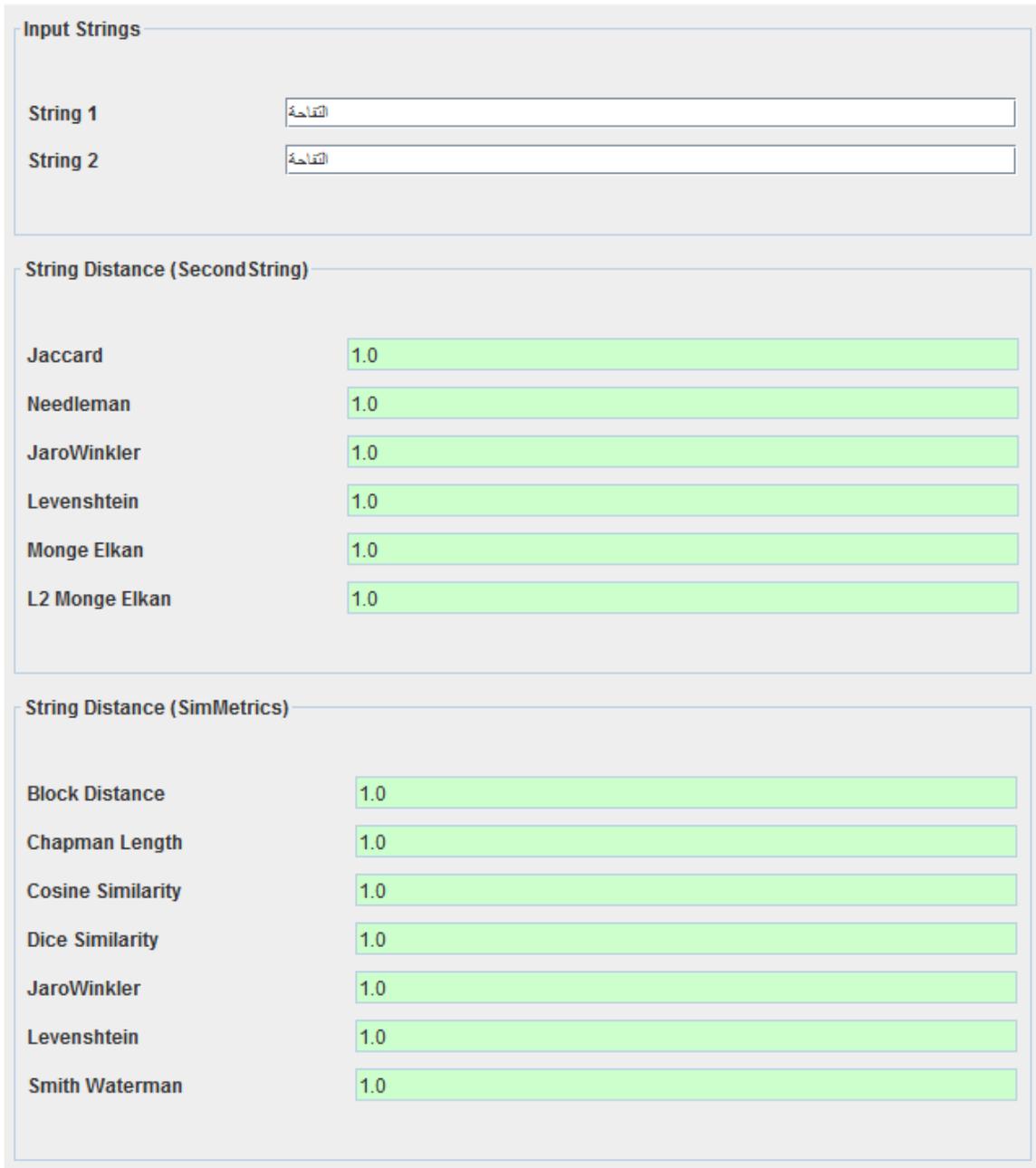


Figure 58 String Matching for Two Exact Strings

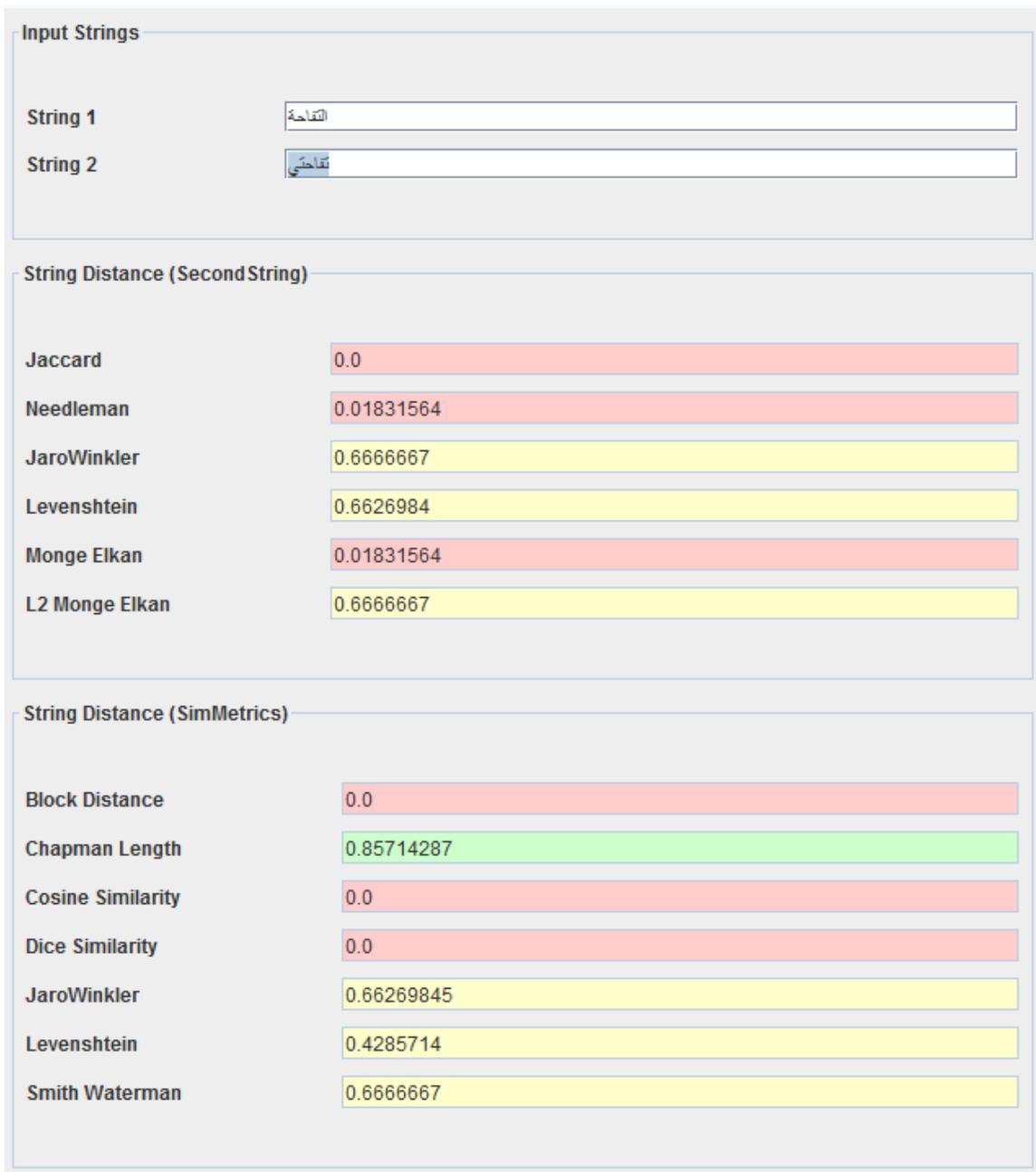


Figure 59 String-Matching Experiment for First Term

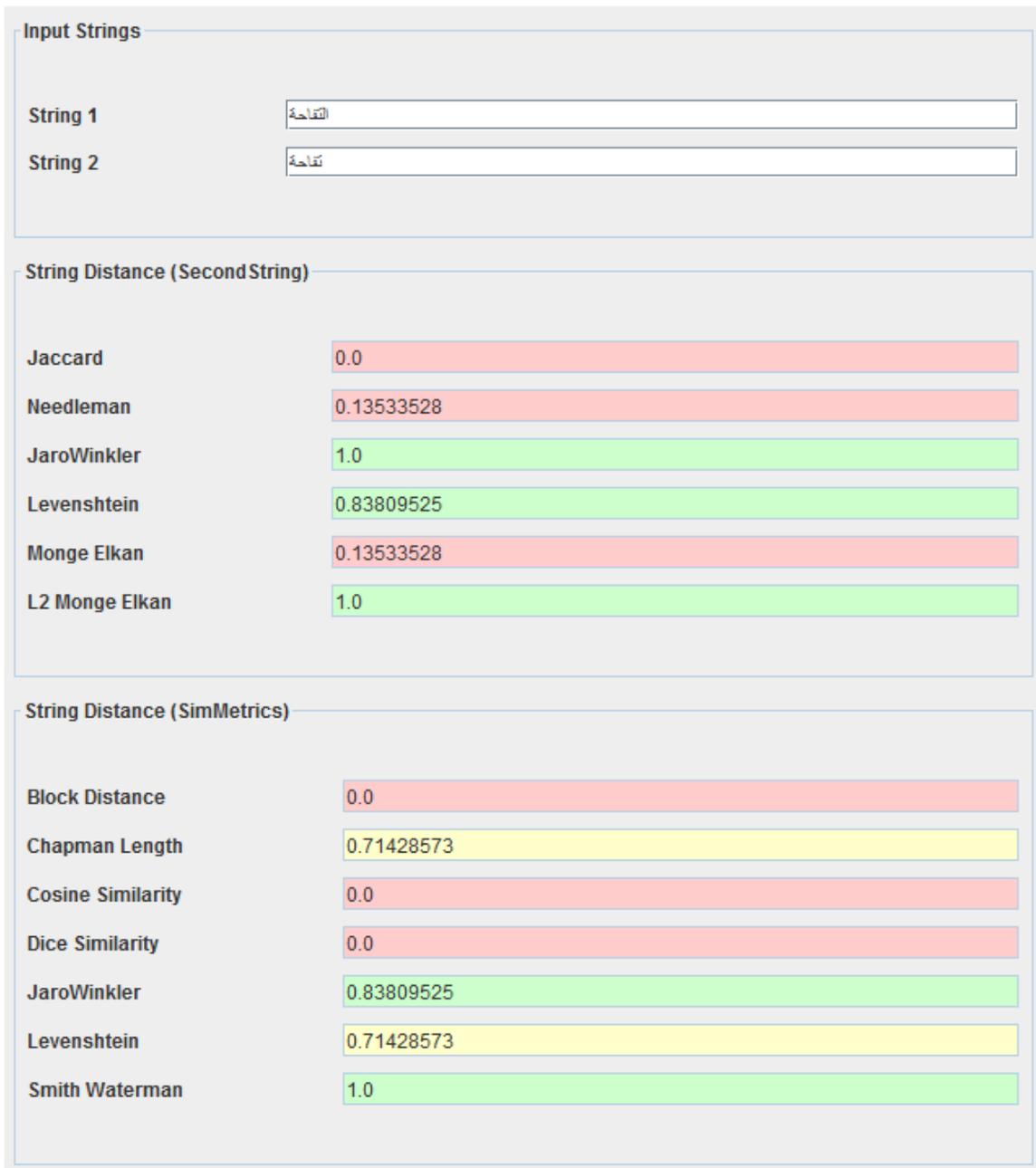


Figure 60 String-Matching Experiment for Second Term

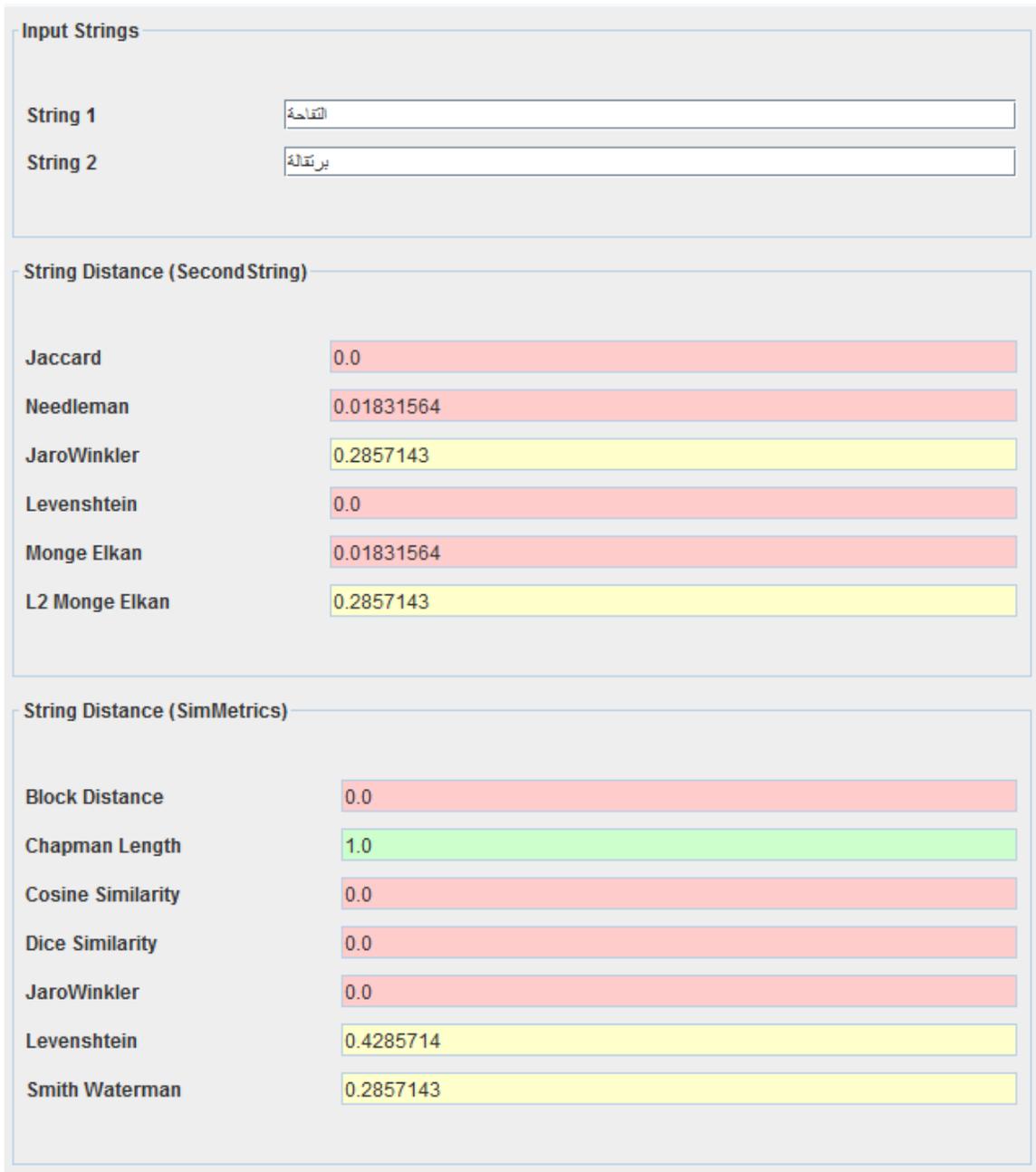


Figure 61 String-Matching Experiment for Third Term

9.3 Query Understanding Experiment

Responses to a total of 453 questions were collected from various sources, such as domain experts, users via surveys, and different health consumer websites. The questions were

categorized based on the existing terms related to the health and food domain ontologies.

TABLE 38 shows the classifications of these questions.

TABLE 38 Question Classifications

Question category	Question type			Total per category
	<i>Yes/No</i>	<i>List</i>	<i>Quantity</i>	
Food centric	37	59	19	115
Nutrition centric	34	31	22	87
Recipe centric	21	27	16	64
Disease centric	29	37	19	85
Body part centric	23	15	9	47
Body function centric	28	19	8	55
Total	172	188	93	453

Figure 62 shows the distribution of the questions based on their category.

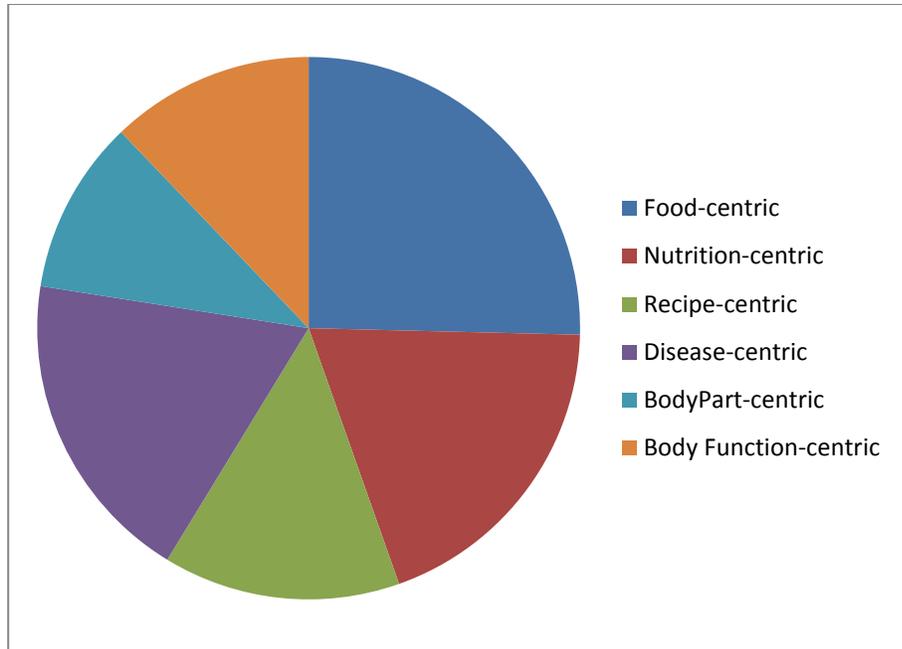


Figure 62 Distribution of the Question Categories

Figure 63 shows the distribution of the questions based on the question category and question type. Most of the list questions are food centric.

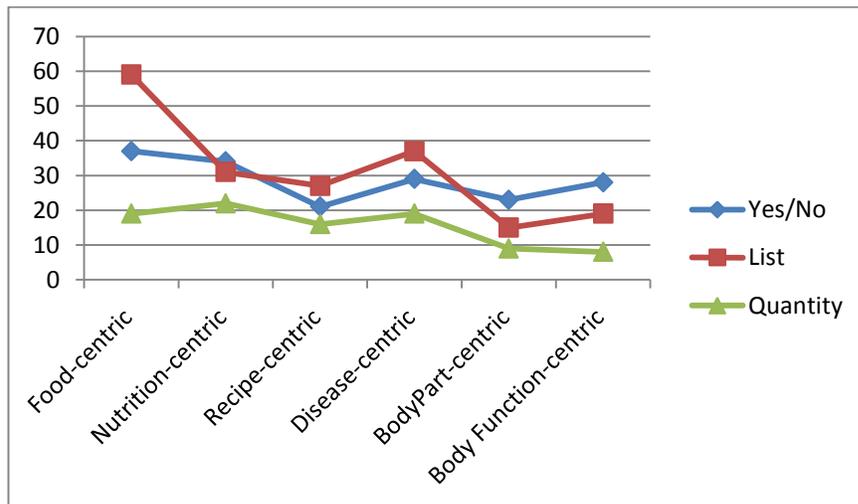


Figure 63 Distribution of the Questions Based on Category and Type

The first experiment is to measure the understanding of the questions semantically. For this, we have manually annotated the questions to identify the number of the related terms to the food and health domains in the questions. We use this for measuring the performance of the question understanding. The performance is measured by precision, recall, and F-measure. The precision measures the accuracy of the results and can be calculated by dividing the correct identified terms by the total of correct and incorrect ones. The recall measures the coverage of the understanding and can be calculated by dividing the correct identified terms by the total terms found manually. F-measure can be calculated using the following equation (142):

$$F = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$$

We show the average measures in the performance chart in Figure 64.

Figure 65 shows the precision, recall, and F-measure line chart across all questions' categories. We observe that they are related and that the more precision we have the higher the recall and then the higher the F-measure value.

TABLE 39 shows the performance of the question understanding. The results show a high precision of 90%, which is required in such a domain to get an accurate understanding of

the questions. The coverage is 76%, which leads to a need to get more knowledgebases for the domain ontologies with richer concepts.

We show the average measures in the performance chart in Figure 64.

Figure 65 shows the precision, recall, and F-measure line chart across all questions' categories. We observe that they are related and that the more precision we have the higher the recall and then the higher the F-measure value.

TABLE 39 Questions Understanding Performance

Question category	Measure		
	<i>Precision</i>	<i>Recall</i>	<i>F-measure</i>
Food centric	0.92	0.86	0.89
Nutrition centric	0.90	0.78	0.84
Recipe centric	0.84	0.64	0.73
Disease centric	0.91	0.77	0.83
Body part centric	0.91	0.81	0.86
Body function centric	0.91	0.72	0.80

Question category	Measure		
	<i>Precision</i>	<i>Recall</i>	<i>F-measure</i>
Average	0.90	0.76	0.83

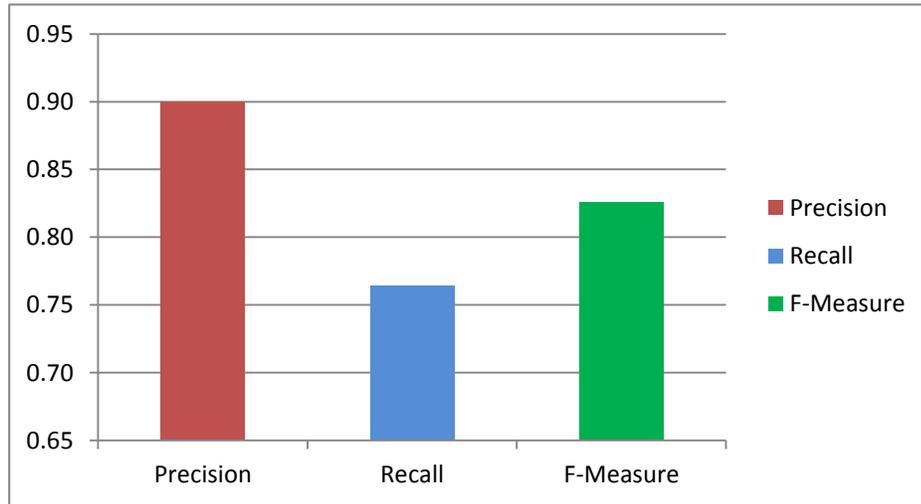


Figure 64 Performance Chart

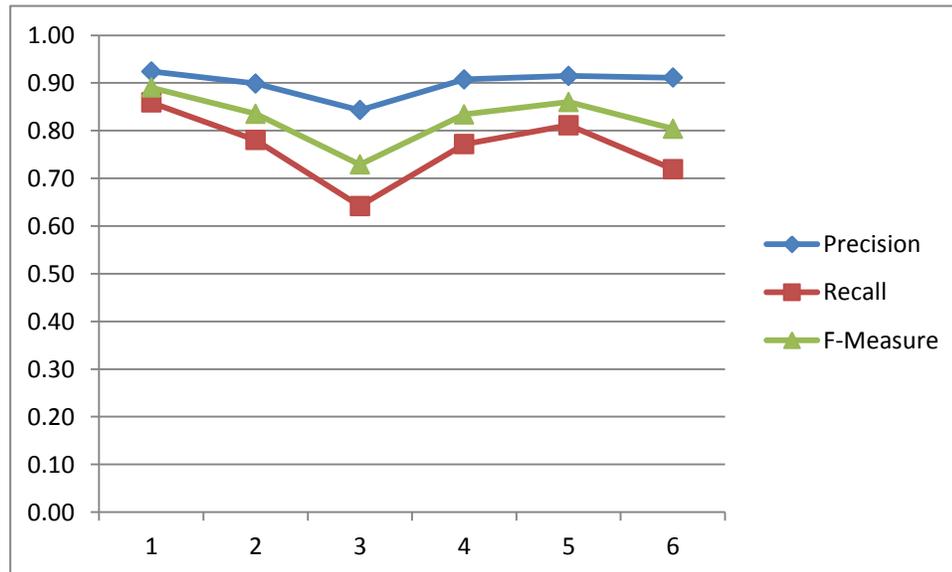


Figure 65 Precision, Recall, and F-Measure Line Chart

9.4 Multilingual Retrieval Experiment

The second experiment is to measure the multilingual capability to retrieve data that were annotated in a different language from the user query language. A set of a hundred Arabic questions was processed, and the retrieved results were checked manually. TABLE 40 shows the performance of the cross-lingual retrieval for each question category. The results also show a high precision of 86%, which means that we could retrieve a good percentage of multilingual results from different sources by understanding the user's question in any language.

TABLE 40 Performance of Cross-Lingual Retrieval

Question category	Number of questions	Total results	Valid results	Precision
Food centric	28	290	245	0.84
Nutrition centric	19	187	164	0.88
Recipe centric	13	195	174	0.89
Disease centric	21	214	192	0.90
Body part centric	8	90	73	0.81
Body function centric	11	112	96	0.86
Total	100	Average		0.86

Figure 66 shows the performance of cross-lingual questions in understanding. It shows the highest precision in the disease-centric questions and the lowest in the body parts-centric questions.

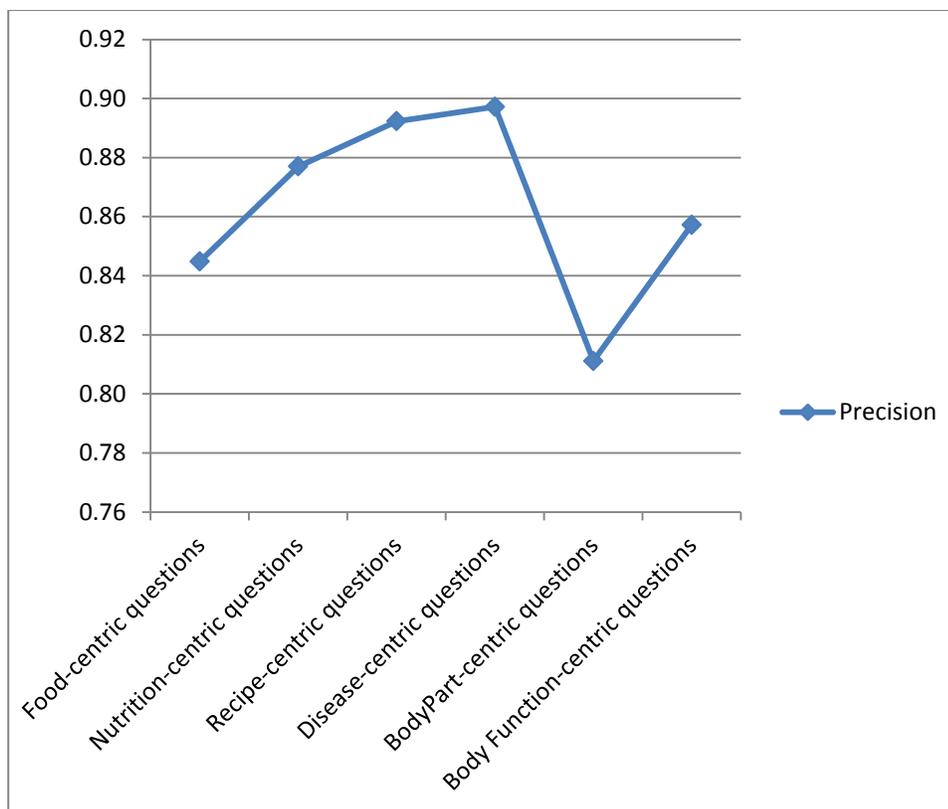


Figure 66 Performance of Cross-Lingual Questions

9.5 Query Enrichment Experiment

The third experiment is to test the questions' semantic enrichment by creating multiple profiles with different values. The enrichment is then done manually and compared with the enrichment done by the system. First, we measure if the system can enrich the question in the expected way that matches the user's profile. Then we measure the satisfaction by getting more relevant results if questions are enriched.

Initial results are promising. Here is an example of the question enrichment. We submitted two questions and retrieved the results with enrichment and without enrichment. We submitted the query, "List the food that has positive impact on diabetes." In the first case where no enrichment is involved, the result is generic and fits any user. In

the second case where enrichment is used, milk is not recommended while tomato is recommended for the user because the user's profile shows an allergy to milk, no preferences for some fruits, such as apples and oranges, and preferences for vegetables. The fact that the query enrichment is subjective motivates us to publicize our work and collect more feedback targeting the individual's preferences.

9.6 Advantages of Semantic Query Manipulation Experiment

The objective of this experiment is to investigate the additional value of the proposed semantic manipulation of the query over the keyword-based traditional information retrieval system.

9.6.1 Annotated Documents Dataset

We have crawled 102,537 documents scattered through 96 trusted websites. Out of these documents, we have selected the richest documents that contain integrated information about all the domains of food, nutrition, and health. These documents have at least two semantic relations between concepts belonging to the three domains. We found 9,852 documents that have at least two semantic relations, which is equivalent to 9.60% of the crawled documents, and that is because that most of the documents have only concepts related to the three domains without relations. TABLE 41 shows the statistics of the top ten crawled websites.

TABLE 41 Top Ten Crawled Websites

No.	Websites	URL	# Doc
1	US Food and Drug Administration	www.fda.gov	523
2	Centers for Disease Control and Prevention	www.cdc.gov	319
3	Saudi Medical Journal	www.smj.org.sa	1,253
4	Service of the National Library of Medicine	www.pubmed.gov	729
5	New England Journal of Medicine	www.nejm.org	582
6	Medscape Continuing Medical Education	www.medscape.com	356
7	American Medical Association	www.ama-assn.org	3,682
8	American Society of Health System Pharmacists	www.ashp.com	4,253
9	US National Institutes of Health	www.nih.gov	259
10	Arab Center of Nutrition	www.acnut.com	853

TABLE 42 and

Figure 67 show the distribution of the selected 9,852 documents that have at least two semantic relations in terms of how many relations exist in these documents.

TABLE 42 Distribution of Selected Documents Based on Number of Relations

Number of relations	Number of documents
more than 5	157
5	316
4	586
3	4,075
2	4,718

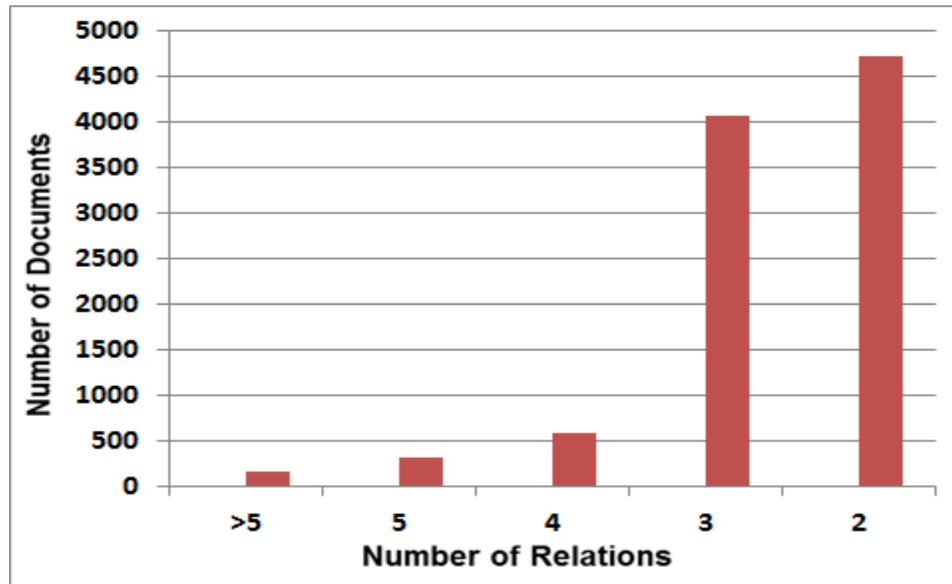


Figure 67 Distribution of Selected Documents Based on Number of Relations

9.6.2 Question Dataset Annotations

We have annotated the 453 questions by two annotators, and TABLE 43 shows the result of the contingency between the two annotations.

TABLE 43 Contingency results for the Two Annotators

		Annotator – 1						Marginal sum
		Food	Nutrition	Disease	Recipe	Body part	Body function	
Annotator - 2	Food	170	4	5	3	1	0	183
	Nutrition	3	40	4	1	3	0	51
	Recipe	1	3	120	2	1	1	128
	Disease	7	6	4	197	5	2	221
	Body part	4	1	1	4	55	1	66
	Body function	5	3	2	1	6	25	42
	Marginal sum	190	57	136	208	71	29	691

We first calculate that the observed percentage agreement is $\text{Pr}(a)$, which is the total number of agreements on different entities divided by the total annotations:

$$\text{Pr}(a) = (170+40+120+197+55+25)/691 = 0.8784$$

Then we calculate the probability of random agreement ($\Pr(e)$), which is the total number of random agreement probabilities for each entity (e.g., $\Pr(e)_{\text{food}}$). For each entity, we calculate its random agreement probabilities by multiplying the total annotated entities of the entity (e.g., food, by each annotator divided by the total annotations). Thus:

- $\Pr(e)_{\text{food}} = 183 / 691 * 190 / 691 = 0.0728$
- $\Pr(e)_{\text{nutrition}} = 51 / 691 * 57 / 691 = 0.0061$
- $\Pr(e)_{\text{recipe}} = 128 / 691 * 136 / 691 = 0.0365$
- $\Pr(e)_{\text{disease}} = 221 / 691 * 208 / 691 = 0.0963$
- $\Pr(e)_{\text{body part}} = 66 / 691 * 71 / 691 = 0.0098$
- $\Pr(e)_{\text{body function}} = 42 / 691 * 29 / 691 = 0.0026$

$$\Pr(e) = 0.0728 + 0.0061 + 0.0365 + 0.0963 + 0.0098 + 0.0026 = 0.2240$$

Then we calculate Cohen's kappa coefficient,³³ which measures the agreement between the two annotations using the formula:

$$\kappa = \frac{\Pr(a) - \Pr(e)}{1 - \Pr(e)},$$

where $k = 0.8784 - 0.2240 / (1 - 0.2240) = 0.8433$.

This means the two annotators agree on 84% of the annotations. The remaining 16% disagreement is due to different interpretation of the questions terms. For example, if there is a question, "What can improve cholesterol?" then the cholesterol might be categorized as nutrition or disease.

³³ http://en.wikipedia.org/wiki/Cohen's_kappa

We notice that the noun phrases in Arabic questions are more numerous than the noun phrases in English questions because of Arabic's characteristics, which use more nouns. The column that shows the percentage from total noun phrases determines the minimum recall percentage if we have random selection and assignment of each noun phrase to a certain category (e.g., the probability of assigning any noun phrase to food category is 19.47%). We should mention that the remaining percentage is 30.47%, which means that the probability of having a noun phrase that could not be assigned to the six categories (food, nutrition, recipe, disease, body part, and body function) is 30.47%. These noun phrases, which are not assigned to the pre-defined categories, might be relations or unknown phrases.

9.6.4 Semantic Query Manipulation with the Traditional Information Retrieval System

The semantic query manipulation adds the following features to the query:

- F1: Missing and implicit terms
- F2: Ontology's vocabulary that matches the user's keyword
- F3: Language representation of the user's native language query

For example, if the user asks, "What food can help diabetes?" then the semantic manipulation of the user's query contains:

- (Disease, is-a, Diabetes)
- (Food?, has-positive-effect, Diabetes)

The semantic query understanding includes the following tasks:

- T1: Named entity recognition: includes virtual term attributes (like color in "green apple")

- T2: Relation recognition: includes injection of missing or implicit terms (virtual term entity)
- T3: Need recognition: includes type of question and type of answer

9.6.5 Experiment Scenarios

In our experiment, we ran three different scenarios:

- 1- Run the user's natural language query as-is in the traditional information retrieval system.
- 2- Enrich the user's query with the additional features (F1, F2, and F3) and then rewrite the user's query with the enrichments. Next, run the enriched query in the traditional information retrieval system.
- 3- Enrich the user's query with the additional features (F1, F2, and F3) and then produce a SPAQL query with the enrichment. Next, run the SPARQL-enriched query on the annotated documents using a semantic reasoner.

9.6.6 Experiment Steps

We have the following steps in our experiment:

- 1- We have run the semantic query manipulation process on the 453 user's queries, which gave us the following:
 - a. Enriched semantic queries;
 - b. Rewritten enriched queries in natural language;
 - c. SPARQL-enriched queries;
 - d. Name entities related to food and health domains;
 - e. Relations related to the domain; and

f. Question type and expected answer type.

- 2- We indexed the 9,852 related documents using the traditional information retrieval system.
- 3- We correlated the 9,852 annotated documents to the 453 queries semi-automatically by finding the documents that contain the identified named entities related to food and health domains (4b) using the traditional information retrieval system with the Boolean relation (AND). We found 2,386 documents related to the 453 queries.
- 4- We annotated the 2,386 documents using the OSAPIR system.
- 5- We ran the 453 queries as-is using the traditional information retrieval system on the 2,386 documents.
- 6- We ran the rewritten enriched 453 queries (4b) using the traditional information retrieval system on the 2,386 documents.
- 7- We ran the SPARQL-enriched queries (4c) using the OSAPIR system on the 2,386 annotated documents.
- 8- We compared the results of step 5 and step 6 to identify the improvements of the semantic query manipulation and enrichment over the traditional query manipulation (keyword-based) by comparing the calculated precision, recall, and F-measure for each.
- 9- We compared the results of step 6 and step 7 to measure the advancement of the OSAPIR system with the semantic query manipulation and enrichment over the traditional system with query enrichment by comparing the calculated precision, recall, and F-measure for each.

9.6.7 Experiment Execution

In this experiment, the idea is to evaluate the advantage of the semantic query manipulation over the traditional information retrieval system and then to compare the OSAPIR system with the traditional information retrieval system, which relies on general search terms or keywords in the document to find the resulting documents for the user query. We select Lucene³⁴ for our experiment as a traditional information retrieval system. The experiment is performed with respect to the relevancy of the results.

9.6.8 Experiment Results

TABLE 45, TABLE 46, and TABLE 47 show the collected results of the experiment from both systems: the Lucene system without query enrichment, the Lucene system with query enrichment, and finally the OSAPIR system with query enrichment.

TABLE 45 Lucene System Performance without Query Enrichment

Measure	Precision	Recall	F-measure
Minimum	0.17	0.29	0.22
Maximum	0.94	1.00	0.97
Average	0.45	0.60	0.50
Standard deviation	0.180	0.194	0.173

³⁴ <http://lucene.apache.org/>

TABLE 46 Lucene System Performance with Query Enrichment

Measure	Precision	Recall	F-measure
Minimum	0.47	0.42	0.44
Maximum	1.00	1.00	1.00
Average	0.65	0.74	0.69
Standard deviation	0.113	0.124	0.107

TABLE 47 OSAPIR System Performance with Query Enrichment

Measure	Precision	Recall	F-measure
Minimum	0.75	0.50	0.60
Maximum	1.00	1.00	1.00
Average	0.90	0.87	0.88
Standard Deviation	0.070	0.114	0.075

9.6.9 Experiment Analysis

The performance of Lucene without query enrichment is low when compared to the same experiment after submitting rewritten queries based on the semantic query manipulation. Both precision and recall were improved with the query enrichment. Moreover, the OSAPIR system outperforms Lucene even with query enrichment because the semantic annotations better correspond to the semantic query manipulation. For example, adding the implicit relationship between terms improves the precision and recall of the OSAPIR system. Moreover, Lucene achieves 100% recall for five queries, while OSAPIR achieves twenty-three queries. OSAPIR also achieves twenty-four queries with 100% precision, while Lucene did not reach 100% precision for all queries.

CHAPTER 10

CONCLUSION AND FUTURE WORK

In this thesis, we surveyed state-of-the-art methodologies in capturing users' preferences. Then we proposed a methodology for identifying and capturing the user's personal and cultural preferences, health conditions, and religious constraints related to the food and health domains. In addition, we studied the current research in users' profile representation. We proposed an ontology-based user's profile to represent the user's preferences, and we built multilingual integrated health and food ontologies and the knowledgebase required for semantic query manipulation. Then we integrated the ontology-based user's profile with the domain ontologies to retrieve precise results. In this thesis, we investigated the personalization methodologies that help customize the retrieved results to match the user's needs. We utilized the user's profile ontology to personalize the retrieved health and food information from the knowledgebase. We proposed in this thesis a multilingual agent-based framework for semantic query manipulation and result personalization, namely the agent-based-framework for semantic-query-manipulation and personalized information retrieval (ASPIR). We also modeled the processes in the framework for semantic query manipulation and personalization. We have implemented and evaluated the framework and the results show high precision and promising results with superior user satisfaction.

As a future work, we can capitalize on the power of agents that can be proactive and provide recommendation and advice to users without asking or querying. The agent can sense the time of the day and the weather in addition to the location and then advice the

user with appropriate recommendations. Furthermore, collaborative-based filtering is a future direction, and it requires publicity of the developed work. There are trade-offs between keeping the health and food information private and sharing it with other users. Another future direction is to test the framework on other domains to validate its scalability. Finally, we would work on publicizing the framework to help the community with better health and nutrition advice.

References

- [1] “Total number of Websites & Size of the Internet as of 2013,” *Facts Hunt*. [Online]. Available: <http://www.factshunt.com/2014/01/total-number-of-websites-size-of.html>. [Accessed: 05-May-2014].
- [2] “The world in 2013, ICT facts and figures,” *ITU*. [Online]. Available: <http://www.itu.int/en/ITU-D/Statistics/Documents/facts/ICTFactsFigures2013-e.pdf>. [Accessed: 05-May-2014].
- [3] E. Hyvönen, K. Viljanen, and O. Suominen, “HealthFinland —Finnish Health Information on the Semantic Web,” *Semant. Web*, vol. 4825, pp. 778–791, 2007.
- [4] D. Matsumoto and L. Juang, *Culture and Psychology*, 5th editio. United States: Cengage Learning, Inc., 2012.
- [5] G. Pandey, “The Semantic Web : An Introduction and Issues,” vol. 2, no. 1, pp. 780–786, 2012.
- [6] Mayo Clinic Staff, “Heart-healthy diet: 8 steps to prevent heart disease.” [Online]. Available: <http://www.mayoclinic.org/diseases-conditions/heart-disease/in-depth/heart-healthy-diet/art-20047702>. [Accessed: 05-May-2014].
- [7] L. Hirschman and R. Gaizauskas, “Natural language question answering: the view from here,” *Nat. Lang. Eng.*, vol. 7, no. 04, pp. 275–300, Feb. 2002.
- [8] V. a. Lapshin, “Question-answering systems: Development and prospects,” *Autom. Doc. Math. Linguist.*, vol. 46, no. 3, pp. 138–145, Aug. 2012.
- [9] M. Ramprasath and S. Hariharan, “A Survey on Question Answering System,” *J. Res.*, vol. 2, no. 1, pp. 171–179, 2012.
- [10] E. Hovy, L. Gerber, and U. Hermjakob, “Toward semantics-based answer pinpointing,” *Proc. first Int. Conf. Hum. Lang. Technol. Res.*, pp. 1–7, 2001.
- [11] S. Faro and T. Lecroq, “The Exact Online String Matching Problem: A Review of the Most Recent Results,” *ACM Comput. Surv.*, vol. 45, no. 2, pp. 1–42, Feb. 2013.
- [12] D. Farmakiotou and V. Karkaletsis, “Rule-based named entity recognition for Greek financial texts,” *Proc. Work. Comput. Lexicogr. Multimed. Dictionaries (COMLEX 2000)*, vol. 75–78, 2000.
- [13] F. Graliński and K. Jassem, “Named Entity Recognition in Machine Anonymization,” *Kłopotek, M.A., Przepiorkowski, A., Wierzchoń, A.T., Trojanowski, K. Recent Adv. Intell. Inf. Syst.*, no. 003, pp. 247–260, 2009.

- [14] A. Mikheev, M. Moens, and C. Grover, “Named entity recognition without gazetteers,” *Proc. ninth Conf. Eur. chapter Assoc. Comput. Linguist.*, 1999.
- [15] G. Altenbek, “Rule-based Person Name Recognition for Xinjiang Minority Languages,” *J. Chinese Lang. Comput.*, vol. 15, no. 4, pp. 219–225, 2005.
- [16] G. Adongbieke, “Rule-based Person-name Recognition for Uighur Texts,” *colips.org*, pp. 1–4.
- [17] F. Dalkılıç, S. Gelişli, and B. Diri, “Named Entity Recognition from Turkish texts,” *Signal Process. Commun. Appl. Conf.*, pp. 918–920, 2010.
- [18] K. Riaz, “Rule-based named entity recognition in Urdu,” *Proc. 2010 Named Entities Work.*, no. July, pp. 126–135, 2010.
- [19] S. Abdallah, K. Shaalan, and M. Shoaib, “Integrating Rule-Based System with Classification for Arabic Named Entity Recognition,” pp. 311–322, 2012.
- [20] D. Nadeau and S. Sekine, “A survey of named entity recognition and classification,” no. 1991, pp. 1–20, 2006.
- [21] D. Bikel and S. Miller, “Nymble: a high-performance learning name-finder,” *Proc. Fifth Appl. Nat. Lang. Process. Conf.*, 1997.
- [22] S. Sekine, “NYU: Description of the Japanese NE system used for MET-2,” *Proc. Messag. Underst. Conf.*, 1998.
- [23] A. Borthwick and J. Sterling, “NYU: Description of the MENE named entity system as used in MUC-7,” ... *Conf. (MUC-7)*, 1998.
- [24] M. Asahara and Y. Matsumoto, “Japanese named entity extraction with redundant morphological analysis,” *Proc. Hum. Lang. Technol. Conf. - North Am. chapter Assoc. Comput. Linguist.*, vol. 1, no. June, pp. 8–15, 2003.
- [25] A. McCallum and W. Li, “Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons,” *Proc. seventh Conf. Nat. Lang. Learn. HLT-NAACL 2003*, pp. 188–191, 2003.
- [26] D. Nadeau, P. Turney, and S. Matwin, “Unsupervised named-entity recognition: Generating gazetteers and resolving ambiguity,” pp. 1–12, 2006.
- [27] E. Alfonseca and S. Manandhar, “An unsupervised method for general named entity recognition and automated concept discovery,” *1st Conf. Gen. WordNet*, 2002.

- [28] R. Evans and S. Street, “A framework for named entity recognition in the open domain,” *Proc. Recent Adv. Nat. Lang. Process.*, pp. 137–144, 2003.
- [29] Y. Shinyama and S. Sekine, “Named entity discovery using comparable news articles,” *Proc. 20th Int. Conf. Comput. Linguist.*, p. 848, 2004.
- [30] O. Etzioni, M. Cafarella, D. Downey, A.-M. Popescu, T. Shaked, S. Soderland, D. S. Weld, and A. Yates, “Unsupervised named-entity extraction from the Web: An experimental study,” *Artif. Intell.*, vol. 165, no. 1, pp. 91–134, Jun. 2005.
- [31] S. Kruk, K. Samp, C. O’Nuallain, and B. Davis, “Search interface based on natural language query templates,” *Proc. IADIS*, 2006.
- [32] D. Cantone and S. Faro, “Fast-search algorithms: New efficient variants of the Boyer-Moore pattern-matching algorithm,” *J. Autom. Lang. ...*, 2005.
- [33] M. Barla, M. Tvarožek, and M. Bieliková, “Rule-based user characteristics acquisition from logs with semantics for personalized web-based systems,” *Comput. Informatics*, vol. 28, pp. 1001–1028, 2012.
- [34] F. Carmagnola, F. Cena, and C. Gena, “User model interoperability: a survey,” *User Model. User-adapt. Interact.*, vol. 21, no. 3, pp. 285–331, Feb. 2011.
- [35] S. Gauch and J. Chaffee, “Ontology-based personalized search and browsing,” *Web Intell. Agent Syst.*, vol. 0225676, pp. 1–35, 2003.
- [36] S. Middleton and N. Shadbolt, “Ontological user profiling in recommender systems,” *Inf. Syst.* (, 2004.
- [37] X. and A.-H. T. Jiang, “Ontosearch: A full-text search engine for the semantic web,” *Natl. Conf. Artif.*, 2006.
- [38] M. R. Ghorab, D. Zhou, A. O’Connor, and V. Wade, “Personalised Information Retrieval: survey and classification,” *User Model. User-adapt. Interact.*, vol. 23, no. 4, pp. 381–443, May 2012.
- [39] D. Billsus and M. J. Pazzani, “Adaptive News Access,” pp. 550–570.
- [40] R. Baeza-Yates and B. Ribeiro-Neto, *Modern Information Retrieval: The Concepts and Technology Behind Search*, 2nd editio. Addison-Wesley, 2011.
- [41] M. Speretta, “Personalizing Search Based on User Search Histories,” pp. 1–6, 2004.

- [42] X. Jiang and A.-H. Tan, “Learning and inferencing in user ontology for personalized Semantic Web search,” *Inf. Sci. (Ny)*, vol. 179, no. 16, pp. 2794–2808, Jul. 2009.
- [43] S. Stamou and A. Ntoulas, “Search personalization through query and page topical analysis,” *User Model. User-adapt. Interact.*, vol. 19, no. 1–2, pp. 5–33, Sep. 2008.
- [44] W. Gao, C. Niu, J.-Y. Nie, M. Zhou, J. Hu, K.-F. Wong, and H.-W. Hon, “Cross-lingual query suggestion using query logs of different languages,” *Proc. 30th Annu. Int. ACM SIGIR Conf. Res. Dev. Inf. Retr. - SIGIR '07*, p. 463, 2007.
- [45] B. Smyth and E. Balfe, “Anonymous personalization in collaborative web search,” *Inf. Retr. Boston.*, vol. 9, no. 2, pp. 165–190, Mar. 2006.
- [46] A. Micarelli and F. Sciarrone, “Anatomy and Empirical Evaluation of an Adaptive Web-Based Information Filtering System,” *User Model. User-adapt. Interact.*, vol. 14, no. 2/3, pp. 159–200, Jun. 2004.
- [47] L. Chen and K. Sycara, “WebMate : A Personal Agent for Browsing and Searching *,” pp. 132–139, 1997.
- [48] R. K. Shukla, “Existing Trends and Techniques for Web Personalization,” vol. 9, no. 4, pp. 430–439, 2012.
- [49] I. Psarras and J. Jose, “A System for Adaptive Information Retrieval,” pp. 313–317, 2006.
- [50] James Pitkow, H. Schütze, T. Cass, R. Cooley, D. Turnbull, A. Edmonds, E. Adar, and T. Breuel, “Personallized Search,” *Commun. ACM*, vol. 45, no. 9, pp. 50–55, Sep. 2002.
- [51] X. Tao and Y. Li, “A personalized ontology model for web information gathering,” *Knowl. Data Eng.*, vol. 23, no. 4, pp. 496–511, 2011.
- [52] J. Budzik and K. J. Hammond, “User interactions with everyday applications as context for just-in-time information access,” *Proc. 5th Int. Conf. Intell. user interfaces - IUI '00*, pp. 44–51, 2000.
- [53] E. Agichtein, E. Brill, and S. Dumais, “Improving web search ranking by incorporating user behavior information,” *Proc. 29th Annu. Int. ACM SIGIR Conf. Res. Dev. Inf. Retr. - SIGIR '06*, p. 19, 2006.
- [54] K. Sugiyama, K. Hatano, and M. Yoshikawa, “Adaptive web search based on user profile constructed without any effort from users,” *Proc. 13th Conf. World Wide Web - WWW '04*, p. 675, 2004.

- [55] P.-A. Chirita, C. S. Firan, and W. Nejdl, “Personalized query expansion for the web,” *Proc. 30th Annu. Int. ACM SIGIR Conf. Res. Dev. Inf. Retr. - SIGIR '07*, p. 7, 2007.
- [56] J. Teevan, S. T. Dumais, and E. Horvitz, “Personalizing search via automated analysis of interests and activities,” *Proc. 28th Annu. Int. ACM SIGIR Conf. Res. Dev. Inf. Retr. - SIGIR '05*, p. 449, 2005.
- [57] X. Shen, B. Tan, and C. Zhai, “Implicit user modeling for personalized search,” *Proc. 14th ACM Int. Conf. Inf. Knowl. Manag. - CIKM '05*, p. 824, 2005.
- [58] E. Toch, Y. Wang, and L. F. Cranor, “Personalization and privacy: a survey of privacy risks and remedies in personalization-based systems,” *User Model. User-adapt. Interact.*, vol. 22, no. 1–2, pp. 203–220, Mar. 2012.
- [59] S. S. Anand and B. Mobasher, “Intelligent Techniques for Web Personalization,” vol. 3169, pp. 1–36, 2005.
- [60] Z. K. Malik and C. Fyfe, “Review of Web Personalization,” *J. Emerg. Technol. Web Intell.*, vol. 4, no. 3, pp. 285–296, Aug. 2012.
- [61] S. Aissi and M. S. Gouider, “Personalization in Geographic information systems : A survey.”
- [62] A. Goy, L. Ardissono, G. Petrone, and D. Informatica, “Personalization in E-Commerce Applications,” pp. 1–38.
- [63] “Survey of Research for the Personalized Education Learning Model and Applications for the CMA School of Arts & Sciences,” pp. 1–16, 2010.
- [64] N. Y. Asabere, “A Survey of Personalized Television and Video Recommender Systems and Techniques,” vol. 2, no. 7, pp. 602–608, 2012.
- [65] D. Sontag, R. W. White, K. Collins-thompson, S. Dumais, and P. N. Bennett, “Probabilistic Models for Personalizing Web Search,” *Proc. WSDM 2012*, 2012.
- [66] V. Jain and M. Singh, “Ontology Based Information Retrieval in Semantic Web: A Survey,” *Int. J. Inf. Technol. Comput. Sci.*, vol. 5, no. 10, pp. 62–69, Sep. 2013.
- [67] R. N. Srivastava, D. Bypass, A. Nagar, N. Kumar, A. Nagar, and B. Nagar, “A Survey on use of Evolutionary Techniques in,” vol. 15, no. 42, pp. 94–97, 2014.
- [68] S. Deepa, “A Survey on Information Retrieval Techniques for Mining the Web,” *Rev. Inf.*, 2011.

- [69] A. Sharma, “Intelligent Information Retrieval System : A Survey,” vol. 3, no. 1, pp. 63–70, 2013.
- [70] O. Egozi, S. Markovitch, and E. Gabrilovich, “Concept-Based Information Retrieval using Explicit Semantic Analysis,” vol. 0, no. 0, pp. 1–38, 2008.
- [71] R. Setchi, Q. Tang, and I. Stankov, “Semantic-based information retrieval in support of concept design,” *Adv. Eng. Informatics*, vol. 25, no. 2, pp. 131–146, Apr. 2011.
- [72] B. Koopman, P. Bruza, L. Sitbon, and M. Lawley, “Towards semantic search and inference in electronic medical records: An approach using concept-based information retrieval What this study adds :,” no. 1, pp. 482–488, 2012.
- [73] D. Wimalasuriya and D. Dou, “Ontology-based information extraction: An introduction and a survey of current approaches,” *J. Inf. Sci.*, no. December 2009, pp. 1–20, 2010.
- [74] J. Kang, “An Ontology-Based Recommendation System Using Long-Term and Short-Term Preferences,” *2011 Int. Conf. Inf. Sci. Appl.*, pp. 1–8, Apr. 2011.
- [75] M.-F. Sy, S. Ranwez, J. Montmain, A. Regnault, M. Crampes, and V. Ranwez, “User centered and ontology based information retrieval system for life sciences,” *BMC Bioinformatics*, vol. 13 Suppl 1, no. Suppl 1, p. S4, Jan. 2012.
- [76] M. Fernandez, I. Cantador, V. Lopez, and D. Vallet, “Semantically enhanced Information Retrieval: an ontology-based approach,” *Web Semant. Sci.*, 2011.
- [77] D. Garg and D. Sharma, “Information Retrieval on the Web and its Evaluation,” *Int. J. Comput. Appl.*, vol. 40, no. 3, pp. 26–31, Feb. 2012.
- [78] Z. Liang, Y. Jun, L. Haifeng, and Q. Haibo, “An Improved Ontology-Based User Interest Model,” *Mod. Appl. Sci.*, vol. 6, no. 6, pp. 39–44, May 2012.
- [79] N. Belkin and W. Croft, “Information filtering and information retrieval: two sides of the same coin?,” *Commun. ACM*, no. 12, 1992.
- [80] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to Information Retrieval*. Cambridge University Press, 2008.
- [81] D. Zhou, S. Lawless, and V. Wade, “Improving search via personalized query expansion using social media,” *Inf. Retr. Boston.*, vol. 15, no. 3–4, pp. 218–242, Feb. 2012.

- [82] J. Leveling and G. J. F. Jones, "Classifying and Filtering Blind Feedback Terms to Improve Information Retrieval Effectiveness Categories and Subject Descriptors," 2010.
- [83] P. Ogilvie, E. Voorhees, and J. Callan, "On the number of terms used in automatic query expansion," *Inf. Retr. Boston.*, vol. 12, no. 6, pp. 666–679, Jul. 2009.
- [84] G. Cao, J.-Y. Nie, J. Gao, and S. Robertson, "Selecting good expansion terms for pseudo-relevance feedback," *Proc. 31st Annu. Int. ACM SIGIR Conf. Res. Dev. Inf. Retr. - SIGIR '08*, p. 243, 2008.
- [85] E. W. De Luca and A. Nürnberger, "Adaptive Support for Cross-Language Text Retrieval," pp. 425–429, 2006.
- [86] Z. Yin, M. Shokouhi, and N. Craswell, "Query Expansion Using External Evidence," no. 2, pp. 362–374, 2009.
- [87] H. Cui, J. Wen, J. Nie, and W. Ma, "Query expansion by mining user logs," *IEEE Trans. Knowl. Data Eng.*, vol. 15, no. 4, pp. 829–839, Jul. 2003.
- [88] B. Billerbeck, H. E. Williams, H. I. Storage, and R. Information, "Query Expansion using Associated Queries," pp. 2–9, 2002.
- [89] D. Nguyen, A. Overwijk, C. Hauff, R. B. Trieschnigg, D. Hiemstra, and F. M. G. De Jong, "WikiTranslate: Query Translation for Cross-lingual Information Retrieval using only Wikipedia."
- [90] Y. Song and L. He, "Optimal rare query suggestion with implicit user feedback," *Proc. 19th Int. Conf. World wide web - WWW '10*, p. 901, 2010.
- [91] J. Wu, I. Ilyas, and G. Weddell, "A Study of Ontology-based Query Expansion," 2011.
- [92] D. Vallet and P. Castells, "Personalized content retrieval in context using ontological knowledge," *Syst.*, no. c, pp. 1–9, 2007.
- [93] I. Ruthven and M. Lalmas, "A survey on the use of relevance feedback for information access systems," *Knowl. Eng. Rev.*, vol. 18, no. 2, pp. 95–145, Jun. 2003.
- [94] D. Shen, R. Pan, J.-T. Sun, J. J. Pan, K. Wu, J. Yin, and Q. Yang, "Query enrichment for web-query classification," *ACM Trans. Inf. Syst.*, vol. 24, no. 3, pp. 320–352, Jul. 2006.
- [95] H. Bast, D. Majumdar, and I. Weber, "Efficient interactive query expansion with complete search," ... *Sixt. ACM Conf. ...*, pp. 857–860, 2007.

- [96] J. Ruvini, “Adapting to the user’s internet search strategy,” *User Model. 2003*, pp. 55–64, 2003.
- [97] E. N. Efthimiadis, “Interactive query expansion: A user-based evaluation in a relevance feedback environment,” *J. Am. Soc. Inf. Sci.*, vol. 51, no. 11, pp. 989–1003, 2000.
- [98] S. Z. Saad, “Web Personalization based on Usage Mining,” pp. 15–21.
- [99] R. Van Meteren and M. Van Someren, “Using content-based filtering for recommendation,” ... *Mach. Learn. ...*, 2000.
- [100] M. J. Pazzani and D. Billsus, “Content-Based Recommendation Systems,” pp. 325–341, 2007.
- [101] V. Maidel, P. Shoval, B. Shapira, and M. Taieb-Maimon, “Evaluation of an ontology-content based filtering method for a personalized newspaper,” *Proc. 2008 ACM Conf. Recomm. Syst. - RecSys '08*, p. 91, 2008.
- [102] P. Lops, M. De Gemmis, and G. Semeraro, *Content-based Recommender Systems: State of the Art and Trends*. Boston, MA: Springer US, 2011, pp. 73–105.
- [103] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, “Item-based collaborative filtering recommendation algorithms,” *Proc. 10th ...*, 2001.
- [104] M. Deshpande and G. Karypis, “Item-based top- N recommendation algorithms,” *ACM Trans. Inf. Syst.*, vol. 22, no. 1, pp. 143–177, Jan. 2004.
- [105] M. Jamali and M. Ester, “TrustWalker: a random walk model for combining trust-based and item-based recommendation,” *Proc. 15th ACM SIGKDD Int. ...*, pp. 397–405, 2009.
- [106] G.-R. Xue, C. Lin, Q. Yang, W. Xi, H.-J. Zeng, Y. Yu, and Z. Chen, “Scalable collaborative filtering using cluster-based smoothing,” *Proc. 28th Annu. Int. ACM SIGIR Conf. Res. Dev. Inf. Retr. - SIGIR '05*, p. 114, 2005.
- [107] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, “Recommender systems for large-scale e-commerce: Scalable neighborhood formation using clustering,” *Proc. fifth ...*, 2002.
- [108] T. George and S. Merugu, “A Scalable Collaborative Filtering Framework Based on Co-Clustering,” *Fifth IEEE Int. Conf. Data Min.*, pp. 625–628, 2005.
- [109] M. Baumgarten and A. Büchner, “User-driven navigation pattern discovery from internet data,” *Web Usage Anal. ...*, vol. 26749, pp. 74–91, 2000.

- [110] B. Mobasher, H. Dai, T. Luo, and M. Nakagawa, “Effective personalization based on association rule discovery from web usage data,” *Proc. 3rd Int. Work. Web Inf. data Manag.*, 2001.
- [111] W. Lin, S. Alvarez, and C. Ruiz, “Efficient adaptive-support association rule mining for recommender systems,” *Data Min. Knowl. Discov.*, 2002.
- [112] P.-N. Tan, V. Kumar, and J. Srivastava, “Selecting the right objective measure for association analysis,” *Inf. Syst.*, vol. 29, no. 4, pp. 293–313, Jun. 2004.
- [113] C. Aggarwal, J. Wolf, K. Wu, and P. Yu, “Horting hatches an egg: A new graph-theoretic approach to collaborative filtering,” *Proc. Fifth ACM SIGKDD Int'l Conf. Knowl. Discov. Data Min.*, 1999.
- [114] B. Mirza, B. Keller, and N. Ramakrishnan, “Studying recommendation algorithms by graph analysis,” *J. Intell. Inf.*, 2003.
- [115] R. Burke, “Hybrid web recommender systems,” *Adapt. web*, pp. 377–408, 2007.
- [116] M. Godse, R. Sonar, and A. Jadhav, “A Hybrid Approach for Knowledge-Based Product,” pp. 268–279, 2009.
- [117] M. Nakagawa and B. Mobasher, “A hybrid web personalization model based on site connectivity,” *Proc. of WebKDD*, 2003.
- [118] S. Khonsha and M. H. Sadreddini, “New hybrid web personalization framework,” *2011 IEEE 3rd Int. Conf. Commun. Softw. Networks*, pp. 86–92, May 2011.
- [119] J. Lee, J.-K. Min, and C.-W. Chung, “An effective semantic search technique using ontology,” *Proc. 18th Int. Conf. World wide web - WWW '09*, p. 1057, 2009.
- [120] A. Harth, S. Kinsella, and S. Decker, “Using Naming Authority to Rank Data and Ontologies for Web Search,” pp. 277–292, 2009.
- [121] et al. A. M. Riad, Hamdy K. Elminir, “PSSE: An Architecture For A Personalized Semantic Search Engine,” *Int. J. Adv. Inf. Sci. Serv. Sci.*, vol. 2, no. 1, pp. 102–112, Mar. 2010.
- [122] M. Burgin and G. Dodig-Crnkovic, “A Systematic Approach to Artificial Agents,” *Prepr. Comput. Sci. cs.AI. 0902.3513*, 2009.
- [123] D. J. Barnes and D. Chu, *Agent-Based Modeling*. Springer London, 2010, pp. 21–77.
- [124] “Preamble to the Constitution of the World Health Organization as adopted by the International Health Conference, New York, 19-22 June, 1946; signed on 22 July

- 1946 by the representatives of 61 States.” [Online]. Available: <http://www.who.int/about/definition/en/print.html>.
- [125] “Nutrition.” [Online]. Available: <http://en.wikipedia.org/wiki/Nutrition>. [Accessed: 05-May-2014].
- [126] M. Boguski, “Online health information retrieval by consumers and the challenge of personal genomics,” *Essentials Genomic Pers. Med.*, pp. 252–258, 2008.
- [127] O. Suominen, E. Hyvönen, K. Viljanen, and E. Hukka, “Web Semantics : Science , Services and Agents on the World Wide Web HealthFinland — A national semantic publishing network and portal for health information,” *World Wide Web Internet Web Inf. Syst.*, vol. 7, pp. 287–297, 2009.
- [128] Y. Wang and Z. Liu, “Personalized Health Information Retrieval System.,” *AMIA Annu. Symp. Proc.*, p. 1149, Jan. 2005.
- [129] S. Abidi, “Adaptable personalized care planning via a semantic web framework,” *20th Intl Cong Eur. Fed Med. Informatics*, 2006.
- [130] S. Chessa, E. De Vega, C. Vera, M. T. Arredondo, M. García, A. Blanco, and R. De Heras, “Adaptive Searching Mechanisms for a Cardiology Information Retrieval System,” *Comput. Cardiol.*, pp. 147–150, 2005.
- [131] Y. Li and J. Mostafa, “A Privacy Enhancing Infomediary for Retrieving Personalized Health Information from the Web,” *Pers. Inf. Manag.*, pp. 82–85, 2006.
- [132] S. Sahay and A. Ram, “Socio-Semantic Health Information Access.,” *Proc. AAAI Spring Symp. AI Heal. Commun.*, 2011.
- [133] S. Al-Bukhitan, “Multilingual Framework for Ontology-Based Semantic Annotation of Health, and Nutrition Websites,” King Fahd University of Petroleum and Minerals, 2014.
- [134] A. M. Iqbal, “A Network of Heterogeneous and Distributed Ontologies for Health and Nutrition Information System,” King Fahd University of Petroleum and Minerals, 2014.
- [135] G. Di, M. Serugendo, and M. Gleizes, “Agents and Multi-Agent Systems,” in *Self-organising Software*, Springer, 2011, pp. 105–119.
- [136] D. Dominguez, F. Grasso, T. Miller, and R. Serafin, “PIPS: An Integrated Environment for Health Care Delivery and Healthy Lifestyle Support,” in *4th Workshop on Agent applied in Healthcare, ECAI2006*, 2006.

- [137] M. Thangaraj and M. Chamundeeswari, "A Survey of Agent-based Personalized Semantic information retrieval," *IJCST*, vol. 2, no. 3, pp. 488–498, 2011.
- [138] "Semantic Web." [Online]. Available: <http://www.w3.org/standards/semanticweb/>. [Accessed: 05-May-2014].
- [139] M. D., B. A., M. S., B. J., and K. Y., "A domain specific ontology based semantic web search engine," 2011.
- [140] D. Damjanovic, V. Tablan, and K. Bontcheva, "A Text-based Query Interface to OWL Ontologies.," *LREC*, 2008.
- [141] H. S. Pinto and J. P. Martins, "Ontologies: How can They be Built?," *Knowl. Inf. Syst.*, vol. 6, no. 4, pp. 441–464, Mar. 2004.
- [142] M. Contreras and E. Germán, "Design and implementation of a FIPA compliant agent platform in .NET.," *J. Object Technol*, vol. 3, no. 9, pp. 5–28, 2004.
- [143] "Precision and recall." [Online]. Available: http://en.wikipedia.org/wiki/Precision_and_recall. [Accessed: 12-Feb-2014].

Vitae

Name : Ahmed A. Al-Nazer

Nationality : Saudi

Date of Birth :4/22/1979

Email : alnazera@gmail.com

Address : P.O. Box.20223 Dhahran 31311, Saudi Arabia

Academic Background : BS: Computer Science 2001, MS: Computer Science 2006

I graduated from King Fahd University of Petroleum and Minerals (KFUPM) in 2001 and hold a BS in Computer Science with a second honor. My senior project, E-trip Portal, won an award as one of the best university projects in the 2001 annual exhibition. Since 2001, I have been working in the information technology (IT) department of the largest oil producer company in the world, Saudi Aramco. This work has exposed me to real-world IT deployments. While working for this company, I have also pursued both an MS and a PhD in Computer Science at KFUPM as a part-time student. I completed my master's degree in February 2006. My master's thesis title is "Collaborative Autonomous Interface Agent for Personalized Web Search," and my advisor was Dr. Tarek Helmy. In 2007, I started my PhD study in computer science and engineering. I defended my thesis on May 12, 2014. My research interests include Semantic Web, information retrieval and personalization. I have published number of journal and conference papers in related topics during my graduate studies.