Enhancing Recommendations in Specialist Search Through Semantic-based Techniques and Multiple Resources

Abdullah Nasser S Almuhaimeed

School of Computer Science and Electronic Engineering

University of Essex

A thesis submitted for the degree of

Doctor of Philosophy in Computer Science

September 2016

Abstract

Information resources abound on the Internet, but mining these resources is a non-trivial task. Such abundance has raised the need to enhance services provided to users, such as recommendations. The purpose of this work is to explore how better recommendations can be provided to specialists in specific domains such as bioinformatics by introducing semantic techniques that reason through different resources and using specialist search techniques. Such techniques exploit semantic relations and hidden associations that occur as a result of the information overlapping among various concepts in multiple bioinformatics resources such as ontologies, websites and corpora. Thus, this work introduces a new method that reasons over different bioinformatics resources and then discovers and exploits different relations and information that may not exist in the original resources. Such relations may be discovered as a consequence of the information overlapping, such as the sibling and semantic similarity relations, to enhance the accuracy of the recommendations provided on bioinformatics content (e.g. articles). In addition, this research introduces a set of semantic rules that are able to extract different semantic information and relations inferred among various bioinformatics resources. This project introduces these semantic-based methods as part of a recommendation service within a content-based system. Moreover, it uses specialists' interests to enhance the provided recommendations by employing a method that is collecting user data implicitly. Then, it represents the data as adaptive ontological user profiles for each user based on his/her preferences, which contributes to more accurate recommendations provided to each specialist in the field of bioinformatics.

Dedication

This thesis is dedicated to the memory of my father
Nasser Saad Almuhaimeed
(1942 - 2000)

Acknowledgements

First of all, I would like to express my sincerest thanks to Professor Maria Fasli, who gave me the opportunity to work on this research. Her professional guidance, instructions, comments, valuable advices, criticisms, and patience during my PhD journey, were of utmost importance. Without her professional guidance and help, this work would not have been achieved. Professor Maria, I would like to say I really appreciate you being so supportive. In addition, I would like to express my thanks to my examiners Dr. Aladdin Ayesh and Dr. Martin Colley who provided valuable feedbacks which make this thesis much better.

I am also grateful to King Abdulaziz City for Science and Technology in Saudi Arabia for granting me the opportunity to complete my MSc and PhD degrees and for the school of Computer Science and Electronic Engineering at the University of Essex for supporting my research.

Last but not least, I would like to express my gratitude to my mother, my wife, siblings, mother-in-law, father-in-law, siblings-in-law, sweet daughters and friends for their encouragement, inspiration, and support to complete this work and allow me to focus on my research to satisfy my prospective goal and get my PhD. This doctoral thesis is dedicated to them.

Contents

List of Figures

List of Tables

List of Algorithms

List of Publications

In the course of the PhD studies, the following publications were produced:

- Almuhaimeed, A., & Fasli, M. (2014). Exploiting Different Bioinformatics Resources for Enhancing Content Recommendations. *14th International Conference on Web Engineering (ICWE 2014). 8541,* pp. 558-561. Toulouse, France: Springer.

- Almuhaimeed, A., & Fasli, M. (2014). Exploring and Exploiting Knowledge in Multiple Resources. *Computer Science and Electronic Engineering Conference (CEEC), 2014 6th* (pp. 109-114). Colchester, United Kingdom: IEEE.

- Almuhaimeed, A., & Fasli, M. (2015). Employing Semantic Techniques for Exploiting Knowledge in Multiple Resources. *The 8th Saudi Student Conference (SSC 2015).* London, United Kingdom: London imperial college.

- Almuhaimeed, A., & Fasli, M. (2015). A Semantic Method for Multiple Resources Exploitation. *In the 11th International Conference on Semantic Systems pp(113-120).* Vienna, Austria:ACM.

- Almuhaimeed, A., & Fasli, M. (2016). An Automatic Method for Updating Semantic Concepts. *The 9th Saudi Student Conference (SSC 2016).* Birmingham, United Kingdom: University of Birmingham.

Chapter 1

Introduction

1.1 Overview

The World Wide Web (WWW) has made unprecedented levels of information available which is typically explored through search engines. A search for content is typically conducted through keywords, and then a set of articles or links that are relevant to the submitted keywords will be returned. However, this type of search is unable to fulfil a specialist's needs, such as a bioinformatician, where the domain is more complex and includes specialist terminologies and corpora. There are resources, such as ontologies, that can enhance the recommendations as they encapsulate understanding of the domain. But, the problem is that there may be multiple such semantic-based and other resources, and when used in combination, they can enhance searches and recommendations as they provide for a deeper and richer understanding of the domain. Hence, new techniques are required to support specialist search.

The rapid growth of the WWW has led many researchers to focus their research studies on enhancing website utilisation. Therefore, researchers have developed semantic techniques that contribute to the exploitation of valuable information from different resources, such as ontologies [1], in order to develop and inform user services from various perspectives. An abundance of resources has also appeared in the field of bioinformatics, which makes organising diverse resources and presenting them to users (i.e. bioinformatics researchers) a problem that needs to be addressed by developers. Moreover, there are many problems that can be found in bioinformatics that have not been addressed in other fields. The amount of data in the domain of bioinformatics is extensive and growing, and relations among bioinformatics resources may be diverse in comparison with those of other disciplines, which may increase the unsolved problems in the field of bioinformatics. This lack of solutions may also result from the complexity of bioinformatics or the minority of computer scientists who have a deep knowledge of this area. Regardless the reason, there are sets of unanswered problems in the domain of bioinformatics recommender systems in terms of exploiting semantic relations and hidden associations between different resources and presenting effective recommendations to bioinformatics researchers.

This study will concentrate on exploring different semantic techniques that would be applied in constructing recommender systems that are concerned with providing recommendations for bioinformatics researchers according to their interests. The project aims to exploit the data overlap between multiple bioinformatics resources, such as ontologies, websites and corpora, and how one can reason over them in order to enhance the precision of the provided services for each researcher. It also seeks to discover hidden associations and use semantic relations, such as siblings and semantic similarities between different resources. This is to provide effective and relevant recommendations, which will help bioinformatics researchers find new articles, websites and tools, which may contribute to enriching their knowledge or finding up-to-date resources.

There are several obstacles that may be faced in attempting to use multiple resources. These obstacles could result from several causes, including inconsistencies in ontologies, incoherent structure and other reasons. An objective of this research is to address some of these challenges in order to enhance user services. Furthermore, there are some essential processes that will be considered in providing effective recommendations such as mapping and reasoning between diverse resources and exploiting adaptive user profiles. The following example embodies a sample of many cases that our prototype recommender system will perform in extracting semantic relations between different bioinformatics resources such as ontologies.

This example can show how the recommender system will exploit different resources by employing their semantic relations to provide effective recommendations for bioinformatics researchers. Suppose Tom, a bioinformatics scientist, is interested in a concept called cytosolic creatine kinase complex, BM-type, where this concept exists in more than one of our resources. We have three resources: a protein ontology (PO)[1], a gene ontology (GO)[2] and the Wikipedia[3] corpus. Typical systems that do not have multiple resources, such as ontologies in a combined way, will search in their corpora without inferring any relation that could be found between Tom's query or preferences and the articles that exist there. This will leave him unaware of relations that could exist between different concepts in multiple ontologies. The task of our recommender system is to exploit all semantic relations and hidden associations among different resources by performing certain reasoning processes. By doing so, it can provide more relevant recommendations, helping Tom satisfy his search request. The concept of interest to Tom can be found in all of the aforementioned resources with some minor differences in presentation (because each resource has its own way of providing information about the concept). Therefore, if we look at the first ontology (PO)[1], we can find the concept cytosolic creatine kinase complex, BM-type 1 identified by PR:000027247 and described as "a cytosolic creatine kinase complex that is a heterodimer of a B-type subunit and an M-type subunit [PRO:DAN, PMID:8430764]"[1]. This has two types of relations that are related to "GO:0002186 : cytosolic creatine kinase complex"[1] that we can use to extract the relation between two concepts in the ontologies and is related with "PR:000027159: creatine kinase B-type isoform"[1] and "PR:000027157: creatine kinase M-type isaoform"[1].

The other ontology (GO)[2] identifies the concept's parent as GO:0002186 and describes it as "a dimeric protein complex having creatine kinase activity"[2]. It has two types of relations, which are "GO:0002185: A protein complex having creatine kinase activity"[2] and "GO:0044445: Any constituent part of cytosol, that part of the cytoplasm that does not contain membranous or particulate subcellular components"[2]. Based on this information, we can infer that there is a relation that can be found between these ontologies. In addition, the Wikipedia corpus

[1] http://pir.georgetown.edu/pro/

[2] http://geneontology.org/

[3] http://www.wikipedia.org/

describes the concept as Creatine kinase (CK), also known as creatine phosphokinase (CPK) or phospho-creatine kinase (and sometimes incorrectly as creatinine kinase), an enzyme expressed by various tissues and cell types. Moreover, CK has different types: the cytosolic CK enzymes consist of two subunits, which can be either B (brain type) or M (muscle type). There are, therefore, three different isoenzymes: CK-MM, CK- BB and CK-MB[3]. Thus, Wikipedia represents a complement and reference resource to compensate for any weaknesses in describing a specific concept such as Tom's concept of interest. Moreover, from the previous example, our suggested system will be able to extract hidden relations or (in other words) semantic relations such as the one between PR:000027247 in the protein ontology and GO:0044445 in the gene ontology. This was obtained through the relation between PR:000027247 and GO:0002186, but this information does not appear in a single resource. Thus, when Tom tries to find this information (i.e. the hidden relation) from the gene-ontology website, he will not be able to find any relationship between these terms because this information is not explicitly present there. However, our recommender system will be able to infer such a relationship, which will contribute to enhancing Tom's provided recommendations. It will exploit semantic relations and hidden associations between multiple bioinformatics resources. It will thus provide enhanced recommendations and support a specialist search with information gained from the overlap among different resources. Furthermore, it will have a friendly user interface that allows the specialist to deal with our recommender system easily. Also, he/she will be able to select a specific interest on which the provided recommendations can concentrate. Figure 1.1 illustrates the services that a prototype recommender system would be able to perform to satisfy Tom's need in the example described above.

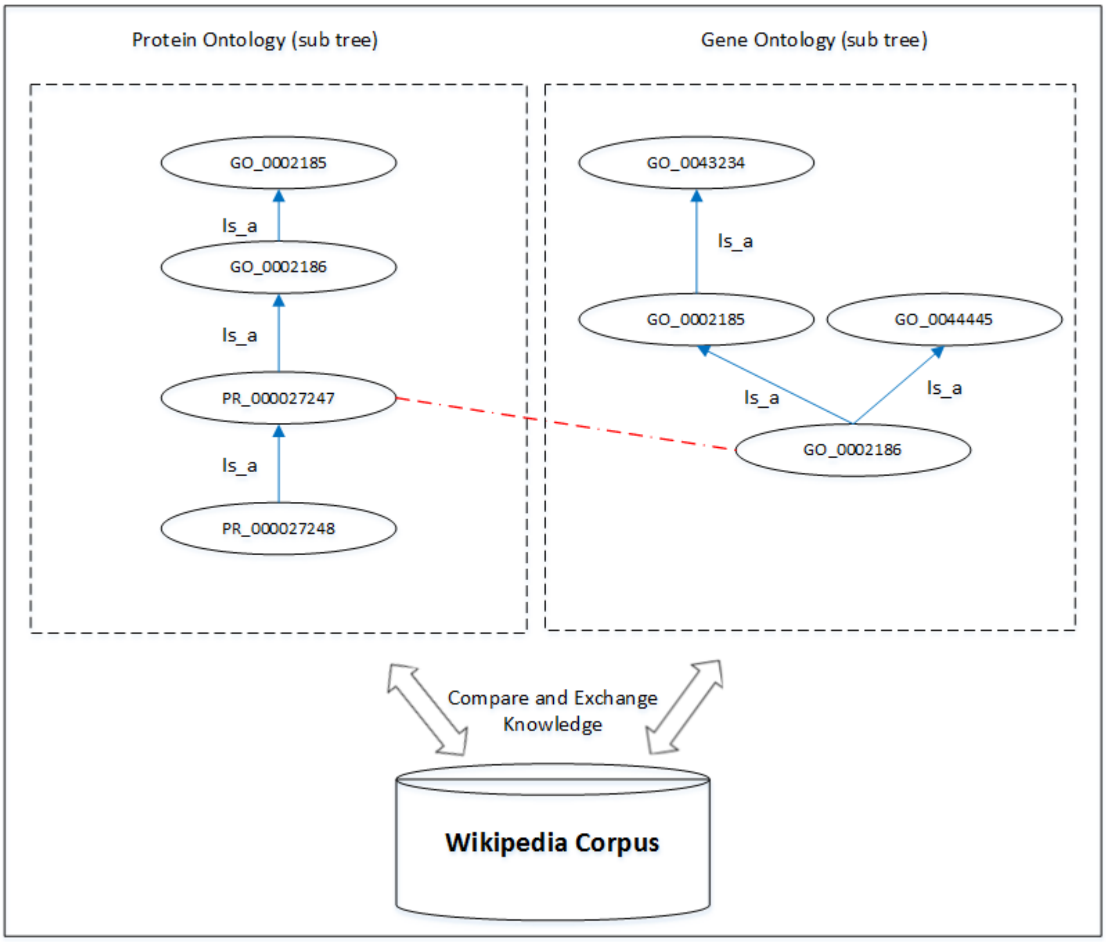


Figure 1.1: Discovering Semantic Relations example.

1.2 Problem Description

In the domain of bioinformatics huge advances has been made and content in the form of specific articles and other information have been steadily increasing. There are many resources in the field of bioinformatics, such as ontologies, databases, websites and corpora. These resources include valuable information about the domain, but each on its own may not be complete, and a combination may also include inconsistent and/or overlapping information. The bioinformatics resources that will be considered in this project include the GO and PO which represent examples of bioinformatics ontologies [2] and [3]. Furthermore, the Open Directory Project (ODP)[1] and Bioinformatics Links Directory (BLD)[2] represent website directories, and the BMC Bioinformatics Corpus (BMC Corpus)[3] contains several bioinformatics articles.

The following scenario presents a bioinformatics scientists problem involving some of the aforementioned resources. This scenario with a bioinformatics scientist will illustrate the main problems that this research will be addressing. Tom is interested in reading content, including articles, webpages and documents, about subcategories in bioinformatics, such as drugs, diseases, genes and proteins. As each of these subcategories might have a huge amount of information, including ontologies, taxonomies, databases and libraries, it would be difficult for Tom to find articles matching all of his preferences. At the same time, the Internet has become the most successful information resource, as it contains a vast amount of information on bioinformatics. The problem is that, in the former case, such data are usually represented as separate ontologies, databases, corpora and libraries, which means that there is no association between such data. On the latter issue, the newly added information on the Internet might not be categorised, so such information might not allow efficient transformation into knowledge that could be used by Tom. Moreover, the frequent changes in Tom's preferences and interests regarding the content he reads may cause another problem that needs to be solved to keep all his preferences updated. This should be done without burdening Tom by asking him to undertake these updates explicitly. There are other factors related to

[1] http://www.dmoz.org/

[2] http://bioinformatics.ca/

[3] http://code.google.com/p/bmc-bioinformatics-processed-corpus/

resource structure or nature, such as ontology inconsistency and ontology structure i.e. some ontologies have Direct Acyclic Graph [4] structures such as GO [5], while others may have hierarchical structures. This variation will complicate the task of providing Tom with effective recommendations relevant to his preferences and interests.

As a result of the aforementioned problems, there is an increasing need for a recommender approach equipped with (i) a semantic-based method that is able to reason through different bioinformatics resources and then extract semantic relations (siblings and semantic similarities) and hidden associations that may occur as a result of information overlapping among different bioinformatics resources. This will allow Tom to find relevant content without consuming his time and effort with this task. Moreover, there is a need for (ii) a method that combines information from multiple resources to improve the understanding of the domain and the precision of the recommendation services. This method works side by side with the semantic-based method and allows it to reason through different resources that have been aggregated together, and it helps it to address inconstancies and incompatibility between different resources. This method supports Tom with diverse knowledge about different topics he prefers without making him search each resource separately and waste time and effort for this purpose. Moreover, this recommender approach should be using (iii) a user profile to provide personalised recommendations that are tailored to Tom's preferences to recommend the most relevant content to him.

Thus, this system should use these tools to provide semantically related recommendations for researchers specialising in the field of bioinformatics. This project will be prepared to recommend new bioinformatics content to researchers (e.g. news, articles, inventions and drugs) related to their preferences and interests. Moreover, the profile should be using techniques that collect user data implicitly to avoid any burden on Tom, so that he is not asked to enter information. Also, the profile should use an automated adaptation method responsible for adding, deleting and updating Tom's preferences over time. Moreover, this user profile should contain a mapping technique that maps new interests with the user's interests and explores semantic information to enhance the provided recommendations. Finally, an extra service could be provided to Tom by offering him the ability to narrow down his interests to specific topics or points included in his user profile.

1.3 Research Aim and Objectives

The aim of this research is to investigate and enhance recommendations on content in the specialist and generic domains; such enhancements will help eliminate the problems described in the previous section. The project's objectives are as follows:

1. To develop novel methods to reason over multiple bioinformatics resources that may contain complementary information to assist the user in specialist searches and enhance the provided recommendations. These methods are applicable in any other domain; however, they will be tested in the bioinformatics domain.

2. To design and implement new techniques to discover hidden associations and infer new semantic relations. These techniques employ the discovered associations and relations (i.e. specifically siblings and semantic similarity relations, since they are the most promising relations among all those discovered) to enhance the precision of recommendations in specialist search.

3. To develop a method that controls drawing/representing all inferred semantic relations and associations. Moreover, it overcomes all difficulties and challenges that result from the inconsistencies between multiple resources with various structures.

4. To evolve a method that provides up-to-date information. This method will work to ensure that our semantic network updated by taking into accounts any changes or updates in the original resources of these inferred data that represent the semantic network. Moreover, these resources should be in a specific format, such as OWL, to complete the update and then perform reasoning on the updated parts.

The work described in this thesis provides detailed answers to the following four questions:

- **What is the main problem that our research intends to address?**
  The main problem of this research is to extract semantic relations and hidden associations between different resources varying in their structures (e.g. ontologies as structured and corpora as unstructured data) and then to exploit them to enhance the precision of recommendations regarding bioinformatics content. So, the overlap between these resources seems to have rich relations that can be employed to enhance the accuracy of the provided recommendations. This will be done for each user individually based on the user's preferences or interests.

- **What is the novelty of this research in comparison to the related approaches?**
  This research can be distinguished from related works by its ability to extract semantic relations (such as sibling and semantic similarity) and hidden associations between different bioinformatics ontologies (e.g. PO, GO, ODP and BLD) and Wikipedia as a corpus in order to exploit such rich information and provide more accurate recommendations for bioinformatics researchers in identifying contents of interest. Moreover, this approach will be fully automated and able to collect users' preferences and interests through their profiles implicitly and to support them with recommendations of semantically related content.

- **What are the steps that will be considered in order to address the main problem as well as its constituent sub-problems?**
  This problem can be addressed by designing a mechanism that reasons between different ontologies and the Wikipedia corpus. Then, it infers and extracts semantic relations (such as sibling and semantic similarity) and associations in order to exploit them for enhancing the accuracy of the recommendations for bioinformatics researchers. Then, a method will be constructed to represent the extracted knowledge, which contains various concepts that have different types of relations, and to exploit it to provide more accurate recommendations. Moreover, an adaptive user profile based on bioinformatics ontology (i.e. ODP bioinformatics branch) will be constructed to assess the use of our approach in general fields. This profile will be equipped with

mechanisms to add, update and delete information automatically without any burden on or intervention from the user.

- **What further desirable features would this project be able to provide?**
  The prototype developed as part of the project will be designed to be user friendly through an interface that gives users the opportunity to interact with the recommender system by selecting a specific concept to narrow recommendations in the selected categories. Moreover, it will allow the user to check recommended items by providing a query which will be considered to enhance the recommendations, but he/she will also be able to get recommendations without submitting any query.

The next chapters will discuss in more detail all the mechanisms and methods that will be developed in our approach in order to satisfy the research aim and objectives.

1.4 Research Contributions

This project will provide content recommender services for specialist domains such as bioinformatics; in other words, the aim is to provide effective recommendations to researchers who specialise in bioinformatics or other disciplines by extracting semantic relations and hidden associations in multiple bioinformatics resources and by considering users' preferences as an essential part of the provided recommendations. This research is also designed to be generic enough to be applied to any other domain besides bioinformatics. This project will make the following contributions:

1. **Semantic-based Method for Specialist Search:** A mechanism is developed to extract new content and semantic relations (e.g. sibling and semantic similarity) between different concepts and hidden associations from different ontologies, such as PO, GO, ODP and BLD as well as the Wikipedia corpus. Then, all of the extracted information is used to enrich the recommendations provided to each user based on his/her preferences or interests. This contribution will fulfil the first aim of this research.

2. **Reasoning Rules and Inference Semantic Relations:** We develop a method with seven semantic rules that are fired during the reasoning process performed over different bioinformatics resources. Then, this method uses our aforementioned semantic-based method in (1) to extract semantic relations and information that satisfy the conditions of any defined rule. This contribution tries to satisfy the second aim of this research.

3. **Method for Representing Semantic Relations and Hidden Associations:** This method is developed to represent rich information, including semantic relations and associations gained from information overlapping between different bioinformatics resources. It will overcome challenges and inconsistencies that appear as a result of multiple resources being combined. Moreover, it will keep the inferred relations and associations up to date based on changes made to the original semantic resources. This contribution is targeted towards fulfilling the third and fourth aims of this work.

4. **Method for Exploiting Semantics between Multiple Resources to Formulate a Semantic Similarity Relation :** This method is developed to perform inference processes between different concepts that occur while formulating the inferred semantic network. It then decides which concepts are semantically similar by considering the semantic similarity between the concepts and the similarity in the concepts' descriptions. This method also addresses the second aim of this research.

Performing these contributions, specifically the semantic-based method (which exploits the inferred relations, i.e. sibling and semantic similarity, to enhance the accuracy of the provided recommendations, and represents the main contribution of this work) will allow this research to fulfil its aim and objectives. Thus, this work will introduce a semantic-based method to reason through different resources with various structures and then extract semantic relations and hidden associations that may be inferred as a result of information overlapping between multiple resources. After that, it represents the discovered and inferred semantic relations and information in a semantic network. The semantic network will be supported by a method that periodically checks if any of the inferred data (i.e. OWL) has changed or been updated to consider the updates in the inferred semantic network. During the reasoning process and while formulating the inferred semantic network, a semantic similarity method will be run to calculate the semantic similarity between different concepts by taking into account the concepts' description similarity and semantic similarity scores. Finally, each user profile will have the most relevant concepts mapped to his/her preferences from the inferred semantic. This will allow the recommendation method to exploit the inferred semantic relations (particularly sibling and semantic similarity, which are the most promising ones) and information to enhance recommendations.

1.5 Structure of the Thesis

The structure of the rest of the thesis is as follows:

- Chapter 2 provides a discussion of the relevant works in the three main parts of the research area, namely (i) the subject of ontologies, which includes a discussion on ontology mapping, semantic similarity and bioinformatics ontologies; (ii) reasoning with multiple resources and reasoning with multiple bioinformatics resources; and (iii) user profiles, which includes modelling, the adaptation of user profiles, recommendations and personalisation approaches and specialist search and recommendations.

- Chapter 3 discusses the contents of the conceptual framework, such as the structure and preparation of resources and components of the user profile. Moreover, it provides a theoretical discussion of our personalised recommendations method. The chapter also discusses the evaluation methods that were used for the recommender systems and the metrics that were used to evaluate our recommender approach.

- Chapter 4 discusses our semantic-based techniques theoretically and shows their methodology from an abstract point of view.

- Chapter 5 provides the implementation methods that were used for all semantic-based techniques and the ontological user profile.

- Chapter 6 provides the evaluation steps that were applied to assess our recommender approach. Also, it discusses the results obtained and compares our recommender system with other recommender approaches, showing the strengths and weaknesses of each approach.
- Chapter 7 provides a critical analysis for all of the developed methods, experiments and results. Also, it discusses the research limitations and introduces possibilities for further work that could enhance our work. Moreover, it contains a comparison between our developed methods and other relevant works.
- Chapter 8 provides the conclusion of the thesis, a list of contributions include references for the aims of this work that have been met in this thesis.
- Appendix A provides snapshots of the recommender search box and retrieved recommendations.
- Appendix B provides snapshots of the plug-in used for collecting user profile data.
- Appendix C provides snapshots of the recommender system service interface.
- Appendix D provides a snapshot of the result-rating interface.
- Appendix E provides a questionnaire and five tasks that were applied to assess the first experiment.
- Appendix F provides a questionnaire and eight tasks that were applied to assess the second experiment.
- Appendix G provides a table represents a comparison between most relevant works, which provides methods that exploit search, semantics, user profile, etc.

Chapter 2

Literature Review

2.1 Overview

There will be several areas discussed within this project, since, it is concerned with exploring semantic techniques that are applied in constructing a recommender system to assist researchers with effective recommendations based on their personal preferences. It is also, interested in the underlying semantic relations as well as hidden associations and information overlapping between several bioinformatics resources. In addition, such a project will support researchers with adaptive user profiles that aid to enhance the efficiency of recommendations based on their preferences. Thus, several topics need to be discussed in order to understand the concepts of ontologies, ontology reasoning, recommendations and user profiles. Therefore, understanding all former topics and prior studies conducted in this field will help us understand the main purpose for this research.

2.2 Ontologies

Ding et al. [6] purport that ontologies are a fundamental concept in the Semantic Web, which is used to represent expert perspectives about a domain in a conceptualised manner. A number of formal languages have been developed to represent ontologies such as the Resource Description Framework (RDF), Resource Description Framework Schema (RDFs) [7] and the Web Ontology Language (OWL) [8].

Formal languages for developing and representing ontologies should contain three essential features: (i) conceptualisation, which involves following a suitable model, such as entity relationships or an object-oriented model, while delivering a consistent ontology that can represent facts; (ii) vocabulary, which covers syntax and grammar; and (iii) axiomatisation, which involves describing the rules and constraints in the language [6]. Cristani and Cuel [9] suggested that an ontology represents the layer that connects the Semantic Web with an information system.

RDF utilises the Uniform Resource Identifiers (URIs) to identify the object of the sentence or triple. This language has been built based on Subject-Predicate-Object (SPO) principle, which means that a single triple is a concept in the ontology [7]. For example, *"Human is a Mammal"* can be represented in RDF as #Human, http://www.w3.org/1999/02/22-rdf-syntax-ns#type,#Mammal, as shown in figure 2.1 "Human" represents a subject in this triple, which takes the form of a source (URI), #type which represents a predicate in this triple, whose form is a source (URI) and "Mammal" which represents an object in this triple, whose form is also a source (URI), or literal text.

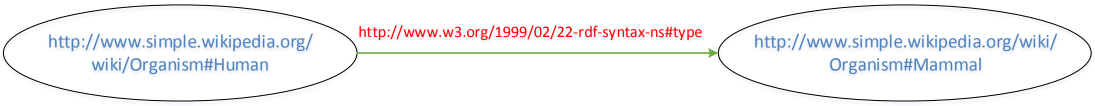

http://www.simple.wikipedia.org/wiki/Organism#Human
http://www.w3.org/1999/02/22-rdf-syntax-ns#type
http://www.simple.wikipedia.org/wiki/Organism#Mammal

Figure 2.1: RDF Example

Although, RDF is used to represent ontologies, it is still weak in reasoning because it is missing many features that may be used and exploited in reasoning such as restrictions and data types [7]. These features or structures can be exploited to infer new relations and information that could exist within processed resources. In contrast, OWL has a more complex structure and is divided into three sub-languages, namely, (i) *OWL-Lite*, which supports users with primary structures that have simple constraints; (ii) *OWL Descriptive Logic (OWL-DL)*, which supports users with maximum information and relations that could be extracted from the ontology and reasoned. OWL-DL can be processed and reasoned by most popular reasoners such as Pellet [10] and HermiT [11]. Finally, (iii) *OWL-Full*, which represents an extension of RDF since it is equipped with OWL and RDF syntax and supports users looking for a fully expressive language. Reasoners are not able to perform reasoning through all components or features that exist in this sub-language [8].

Hartmann et al. [2] illustrated that ontologies help to overcome obstacles such as information overlap and inconsistencies, but this is not an easy task. Ontologies can describe the same domain or slightly different aspects of the domain. So, the semantic overlap between these ontologies can be exploited to enrich their concepts with extra information, which may lead to discovering a new relation or new associations. This also illustrates the importance of using ontologies to describe different resources. In terms of development, Euzenat et al. [12] explained that the development of ontologies imitates software development, which means that an ontology may have several versions, since some researchers or applications update their ontology and others may use old versions of those ontologies.

Furthermore, ontologies have been used in various domains. For instance, Movshovitz-Attias et al. [13] illustrated an answer query system called LATTE that automatically generates sub-concepts and super-concepts to formulate a hierarchy that uses ontologies to answer queries. This system gets most of its power and accuracy from an ontology of attributes from the Web that contains all aspects marked as important by users. As another example of using the ontologies in different domains, Martinez-Cruz et al. [14] developed an ontology that characterises the trust between users by applying a fuzzy model. Thus, recommendations between users will not be based on the similarity of items rated by the user; instead, they are based on users whom he/she trusted before. Also, Cardoso et al. [15] introduced a new architecture for gazetteers by using Volunteered Geographic Information (VGI) with semantic web tool like ontologies to allow gazetteers to overcome their previous problems, such as with handling complex queries. These problems occurred because gazetteers were just using thesauri for names and places, which are limited in handling name disambiguation, unlike other ontologies that help to overcome such problems.

2.2.1 Bioinformatics and Ontologies

The use of ontologies has become increasingly essential in bioinformatics, as ontologies help to overcome research obstacles such as information overlapping and inconsistencies [2]. For example, information inconsistencies between various resources (e.g. databases or corpora) with different types of structures will make extracting rich semantic information difficult. For this reason, ontologies have become essential requirements in classifying bioinformatics data. In addition, ontologies have several uses in the field of bioinformatics, such as annotating and populating ontologies with rich bioinformatics information [16] and exploiting semantic information in ontologies to improve information retrieval and discovery [17], [18] and [19]. For instance, Daraselia et al. [20] suggested an automated mechanism to annotate a gene ontology with rich bioinformatics information using different databases. Another stream of studies focusses on using ontologies and semantic information for neuro-oncological diagnosis. A comprehensive review of such studies can be found in [21]. Another work [22] involved an experiment on ontologies by matching two medical ontologies: Computer Retrieval of Information on Scientific Projects (CRISP[1]) and Medical Subject Headings (MeSH[2]). Foundational Model of Anatomy Ontology (FMA[3]) was used as a reference or background ontology to classify terminologies. In addition, Blondé et al. [23] provided a reasoning approach for Open Biological and Biomedical Ontologies (OBO[4]), which represented a biological reasoning descriptive framework that can be used to infer a great deal of knowledge about a specific product or medicine.

Furthermore, Foulger et al. [24] introduced a project for GO annotation of Parkinson's disease. They discussed stages that should be considered to intensify proteins, publications and cellular processes in annotations. For instance, they discussed how GO annotation can determine information that is relevant to Parkinson's and taking advantage of the approaches that are highly focussed to be provided to the user. Also, Galeota et al. [25] showed the semantic annotations of the Gene Expression Omnibus using meta-data samples of concepts from biomedi-

[1] http://crisp.cit.nih.gov/

[2] http://www.nlm.nih.gov/mesh/

[3] http://sig.biostr.washington.edu/projects/fm/AboutFM.html

[4] http://obofoundry.org/

cal ontologies. In addition, they illustrated how initial queries can be expanded to determine the most semantically similar dataset that can be used for the queries. Moreover, Lekschas and Gehlenborg [26] constructed a system called SATORI, which is an ontology for visual exploration that joins a useful meta-data search with a tree map and a node link diagram that visualise the repository structure. Also, it provides context to retrieve datasets as an interface that allows for semantic query and browsing of the repository. The system's requirements were taken based on biomedical scientists' perspectives to allow this system to address some problems or difficulties that they may face in this area.

All of the aforementioned projects and works show the importance of ontologies in the field of bioinformatics while illustrating the different uses of ontologies for bioinformatics-related applications and problems. Clearly, ontologies are important elements in this research because they represent one of the main resources under consideration.

2.2.2 Ontology Mapping

Ontology mapping can be defined as a collection of compatibilities between different ontologies' elements, in which such compatibilities can be classified using classes, sub-classes, relations and transformation rules [27]. Ontology mapping usually refers to ontology matching, which is the process of incorporating knowledge and information through different ontologies [28]; alternatively, it may refer to ontology alignment, which is the process of conducting links between pairs of ontologies [29] and [27]. Ontology mapping also represents an important process in ontology integration, merging and alignment [29]. By no means is it an easy task, since it requires several subtasks to be undertaken in order to perform ontology mapping, including semantic similarity and matching between ontologies. In addition, there are several problems that may occur as a result of ontology mapping such as inconsistency in some ontologies' semantics or varying configurations in knowledge representation [30]. Thus, as a result of some cases that will need to be mapped in our approach to integrate different resources with each other, a set of relevant works will be highlighted and discussed in this section.

Ontology mapping is widely used in different domains such as e-commerce and e-learning. For instance, Arch-int and Arch-int [31] proposed a semantic mapping method to find compatibilities between different learning resource systems. This method provided expressiveness combinations between triple predicates and elected concepts for the other ontology. Moreover, they suggested a common ontology, such as IEEE LOM[1], to incorporate all known meta data standard for learning. Nuntawong et al. [32] introduced a model that found the correspondences between computer-science courses and the standard of the Thailand Qualifications Framework for Higher Education (TQF:HEd). The aim of their work is to create a curriculum for Thailand universities and decrease time that could be spent searching corresponding courses between different universities. So, in order to reach the desired target they have created an ontology which was connected to a web-based application, which was designed to be able to map a couple of ontologies to find correspondence between them. To make this, they used an extended version of Wu & Palmer's algorithm [33] and WordNet.

Kumar and Harding [34] illustrated a method that performs ontology mapping based on descriptive logic (DL) and bridging the axioms between ontologies. They exploited the atomic concept similarity as an input for the mapping role level and complex concepts. Their method begins by identifying the main parts of the mapping process, such as concepts and roles, by using an ontology editor such as Protégé or Jena API. They then identified the lexical similarities of concepts by using WordNet, which can be used to identify lexical similarities, such as synonyms, to perform such a process. Finally, they employed DL reasoning such as ABox or TBox to infer the bridge facts between different concepts in multiple ontologies that have been mapped to each other via bridging rules. After that, they formulated an ontology that contains all concepts and extracted and inferred relations between these concepts.

Even though this approach is a great effort in the realm of ontology mapping, it is not fully automated. It requires user intervention to complete the mapping process, especially in the last step, which employs DL reasoning to infer relations between different concepts in multiple ontologies. This step requires user assistance or opinion for completion. Thus, this is a limitation in this approach, as people are naturally disparate in their opinions, and this may cause inaccurate

[1] https://ieee-sa.centraldesktop.com/ltsc/

mapping between ontologies. To overcome such a drawback, an automatic component should be integrated to avoid depending on user opinion and to make this approach fully automated.

Hartung et al. [35] also introduced an approach to generate mapping between ontologies by reusing and composing previous mapping with intermediate ontologies. This approach is concerned with the efficiency of composing routes via intermediate ontologies, and it ranks and selects the top-k intermediate composition for mapping ontologies. In this approach, mapping between two ontologies is done by finding a set of correspondences, each with two concepts. Their relatedness is decided by the level of similarity between different correspondences. Therefore, matching algorithm was employed to measure the level of similarity between different correspondences. Thus, the correspondences with higher similarities will be considered as new mapping between ontologies, and this will enrich this approach with new mapping correspondence based on indirect matching, unlike the previous approaches which are based on direct matching.

Although this approach introduced a new way of performing mapping between ontologies, it only considers a similarity with a score of 1 to be a new case of indirect mapping. This represents a limitation, as considering only a score of 1 in the mapping process is still matching the direct cases, and calculating similarity is not sufficient in this case. To overcome this drawback, a threshold should be integrated to measure the level of similarity and consider any mapping with a similarity score lower than the determined threshold.

Knoblock et al. [36] illustrated an approach that allows users to perform mapping between their sources and an existing ontology. They exploit this mapping to generate RDF triples, which can be used for semantic purposes. This approach, called Karma, was designed to perform automatic mapping between the users' sources and the ontology. Moreover, this approach allows users to support the mapping process with some opinions about the level of correctness of mapping between sources and ontology achieved using this approach; this step can overcome any incorrect mapping process. A Karma mapper uses two steps to map sources to the ontology. Firstly, the user connects data to the range of data properties. Secondly, the user extracts paths that occur as a result of the classified relationships between different concepts from both edges in the mapping process. This generates a mapping between the sources and the ontology, and then the final mapping waits for the user's decision.

Although this approach provides useful mapping between an ontology and data sources, it does not provide fully automated services and requires user intervention to perform the mapping process. People have different opinions with regards to the level of correctness of mapping results. Thus, the variety in user opinions and background is a limitation in this approach. To overcome this drawback and have a fully automated approach, the mapping process should not be completely based on user decision; opinion should be optional to enhance the mapping result.

Ehrig and Sure [37] developed an automatic method that maps two ontologies to each other. This process requires some tasks to be performed manually by experts such as determining similarities. Furthermore, the researchers noted that several steps should be followed to map two ontologies. First, they find a pair of nodes in the selected ontologies to be mapped to determine the similarity between them. This method then depends on label similarity, URIs or the same relation; these factors determine the semantic similarity. After this step, rules representing the measurements for similarity were used. They present the overall similarities between the ontologies' units. Determining similarities and applying rules to find the overall similarity are repeated a limited number of times or until the number of changes in each round has an observed fall in its registered value. Finally, low similarity results are deleted, and only the best similarities are shown as a result of ontology mapping.

This method concentrates on the accuracy of ontology matching, since mistakes in matching will cause serious problems [38]. These problems may affect the precision of other processes, such as ontology merging, which may become inefficient or return incorrect results due to low precision in matching the ontologies. Moreover, this method has some rules that decrease the speed of matching between ontologies' elements. For instance, one of this method's rules (R5), which compares between super-concepts in the mapped ontologies, is time-consuming and decreases the mapping speed, since some concepts are mapped directly and there is no need for such rule to be mapped. Thus, to enhance the speed of this method, some rules should be removed or ignored, such as the step or rule used to compare super-concepts, particularly when concepts are mapped directly and do not require further comparison in ontologies' sub-nodes.

Li [39] also proposed an ontology-mapping approach called Lexicon-based Ontology Mapping Tool (LOM), which is based on lexical similarities between ontologies' elements. LOM uses four methods: **whole-term matching, word-constituent matching, synset matching and type matching** [39], in order to undertake the ontology mapping process between different ontologies. Whole-term matching deals with terms as strings [39]. The technique matches each string with others to find exact matches and then returns a 1 to represent a best match or 0 otherwise. Word-constituent matching extracts words in each term, even capital letters and quotation marks. All suffixes or prepositions are not matched as extracted words. Therefore, the extracted words will be compared to find exact matches between words, and the process will return a 1 for a best match and 0 otherwise after all of these matches. Only the match with the highest score will represent the results using this method. The third method, synset matching, deals with the semantic meanings of the extracted words and compares them with WordNet [40]. This matching follows the same steps as the previous method; however, it does not conduct straight matching between the extracted words. Finally, type matching represents the last stage or process used in LOM. This step uses the Suggested Upper Merged Ontology (SUMO) [41] and the Mid-level Ontology (MILO) [42] for proper mapping. LOM takes unmatched words that are returned from the previous steps and matches them with SUMO/MILO. If there is a match, then the best score for matching is confirmed.

However, the LOM method still has some limitations that are discussed by [39]. This researcher points out that the method needs extension to match abbreviations with the original words. There is a shortcoming in the LOM method in mapping some disciplines' terminologies or symbols, such as medicine or chemistry, as LOM cannot distinguish their terminologies. Such disciplines could be introduced to LOM to further improve its use.

Anam et al. [43] introduced a hybrid approach for ontology mapping that provides a fully automated mapping method to perform all mapping processes automatically, without waiting for a developer or user decision to validate the correctness of the mapping process. This approach employs a machine learning algorithm for classifying entities and incremental knowledge acquisition to address matching errors, such as false positives and false negatives at the element level. This approach follows set of steps to complete the mapping process.

i) **Feature Construction :** consists of a set of stages to satisfy its goal. The first step is *extracting the entities of the ontologies*, which is done by retrieving the content of ontologies, such as classes, labels, etc. Next is the *application of text processing techniques*, which includes all text processing such as tokenization, stop-words removal, looking up synonyms, stemming and translation. Then, the *application of string similarity metrics* involves taking the string similarity of the selected data for the previous stages. The similarity score is normalised between 0 and 1, where 0 means no similarity and 1 means similar, and the threshold score increases by 0.1 from 0 based on mapping decision which is true or false, while the ground truth values can be true/false for experts' decisions or opinions, which are provided manually when mapping a couple of classes.

ii) **Element Level Matching** represents the inference stage between the mapped ontologies and consists of a set of steps that need to be performed. *The knowledge base* step represents the rule stage, during which each pair goes through a set of rules to satisfy mapping conditions. *The inference process* step is based on the previous steps decision to start the inferring process. The step follows the censor rule., so when this rule is satisfied, it continues onto the next rule; otherwise, the step stops with a single path and conclusion. *The knowledge acquisition* is the step, that transferring human experts knowledge to a knowledge base system. *The cornerstone cases* step is used to acquire knowledge. Finally, the *validation and verification* step is used to ensure the correctness of the knowledge base added by the experts and to decide on the level of success achieved when consider such knowledge.

iii) **Structure Level Matching** step represents converting the ontologies into a graph structure in order to check the correctness of the matching.

iv) **Aggregation and Extraction of Mappings** is the final step, during which their approach combing mapping founded from structure, element, entities with average, maximum and minimum weights. The final mapping will be chosen based on the consistent means which select the best mapping performance compared to the other mapping methods.

Although this approach provides an accurate mapping method, it suffers from a limitation that restricts its performance. This approach is not fully automated, since it depends on expert or user opinions at some points of the mapping process, specifically in the **Feature Construction** step. Due to the disparity between users' opinions, the mapping will not be quite accurate. To overcome this limitation, a similarity threshold should be considered to decide whether a pair of concepts can be mapped to each other or not.

Khattak et al. [44] suggested a new method of mapping between different dynamic ontologies based on their histories, which change over time. This is done by mentioning all changes that occur within all comparative ontologies in a log file called the History Change Log (HCL) file. Then, the mapping process begins between the comparative pairs, it calls this log file to find any unreliable elements that have occurred as a result of evolving in one of the comparative ontologies. Then, it removes the unreliable element and maps the two ontologies to each other. In case both comparative ontologies witness an evolution, the log file becomes insufficient and they will need a new mapping, instead of using the existed mapping and considering reliable changes that occur over time. This method is useful for large ontologies and resources, since it reduces the time and effort needed to perform the mapping process between different resources.

This method follows valuable way to perform mapping between different resources. However, it still has some limitations and needs enhancements that may lead to better performance. A method should be added to decide whether the mapping process should be re-done between a couple of ontologies, in case both ontologies witness a change or update. Since, some of the updates may not make a major difference in the resources compared to before and re-mapping the two resources may waste time and effort and may result in mistakes that did not occur the first time, especially with large resources such as ACM transactions, Google, etc. Thus, the task of this method is to measure the level of the change or update that happened by setting a threshold for this purpose. Then, if the level of the update exceeds the adjusted threshold, the method will recommend doing the mapping again. Otherwise, it removes unreliable elements and considers the useful updates that occur in both comparative ontologies.

2.2.3 Semantic Similarity

Semantic similarity is often used to describe how similar two terms or concepts are from a semantics point of view (i.e. from a meaning point of view rather than based on syntax). Semantic similarity plays an important role in recommender systems and when attempting to perform semantic-based techniques in any area/discipline.

There are several semantic similarity measurements that depend on factors like the structure of the ontology or the type of relations between different nodes [45]. Blanchard et al. [46] categorised semantic closeness in literature into different types: ***semantic similarity, semantic distance, and semantic relatedness***. Moreover, they defined semantic similarity as a set of significant semantic links between two concepts, such as *is_a* and *part_of*; semantic relatedness is a set of all semantic links between a couple of concepts that reflect the level of closeness between them. Semantic distance is the shortest distance between pairs of concepts. They also classified different semantic measurements in terms of the type of measurements provided. For instance, Resnik [47], [48], [49] and [33] created criteria to measure semantic similarity; Rada et al. [50], [51] and [52] created classifications criteria to measure semantic distance, and Hirst and St-Onge [53] measured semantic relatedness. Furthermore, Li et al.[54] describe that semantic similarity measures between two words can be classified into two types, namely, ***Edge counting based*** (i.e. dictionary) and ***Information theory based*** (i.e. corpus-based). For instance, Rada et al. [50] demonstrated that the minimum number of edges between two concepts can be considered to measure the conceptual distance between the concepts. This approach is considered the edge-counting-based method for calculating semantic similarity between two concepts, where this measure (edge-counting-based) is sufficient with taxonomies such as medical semantic networks. The problem with this measure that it assumes that all links in the taxonomy have the same value, but this not usually true since the value of links may differ from a relation to another. Resnik [47] explained the information-theory-based method as follows: more similarity between two concepts means more shared information between them. Thus, both measures are exclusive on taxonomies and specific type of relations and cannot handle sophisticated structures such as ontologies or semantic networks. There are several measurements designed to measure semantic similarities, taking into account different factors, which will be discussed in this section.

Yang et al. [55] proposed an approach that calculates the semantic similarity between two terms by considering the downward random walk between terms (i.e. genes). They considered two main factors in calculating the semantic similarity between different terms: *the term's ancestors and descendants.* They classified gene annotation into annotated and partially annotated. The former is represented by the term's descendants, as any gene is annotated with the term, and all genes' parents are annotated with that term. The latter involves the gene annotated by parents, not by descendants (partially annotated). This approach is designed to be more flexible to calculate semantic similarity for ontologies in different structures such as hierarchical and Directed Acyclic Graph (DAG) [4], and it is not exclusive to hierarchical ones as in other approaches. This approach has been compared with six different semantic similarity measures to assess its use, and three of these measures are commonly used ([47], [49] and [51]). The remaining are more recently proposed measures: SimUI and SimGIC, proposed in [56] and GraSM([57] and [58]). They applied these measures as well as their proposed measure on the yeast *Saccharomyces cerevisiae*, mRNA co-expression and protein-protein interaction data.

This approach has been designed to be flexible and calculate the semantic similarity between terms in ontologies with different structures. However, it uses the pairwise method, which employs the best match average (BMA) method [56] to annotate a gene, and it has some limitations caused by the combined rules used in the pairwise method. This rule tried to provide a balance in similarity-calculation accuracy, unlike the average rule (AVG) [59] or maximum rule (MAX) [60]. Because the AVG decreases the similarity to less than 1. For instance, if two unrelated genes annotated with the same term, in this method the similarity between them will be 0.5. However, it should be 1 in such a case because they are annotated with the same term. The MAX increases the similarity score between a pair of genes that share some terms to 1. But, it should be less than 1, since they still have different terms that are shared between them. However, the way of calculating semantic similarity in single pairs may affect the pairwise method. To overcome this limitation, this approach should consider the group-wise method [56] for calculating semantic similarity between a set of genes based on an annotated term.

Sánchez et al. [61] investigated a semantic-similarity measure between several ontologies; their evaluations were conducted on biomedical concepts with standard ontologies (WordNet and MeSH). Their semantic-similarity measure consisted of two methods used to overcome the limitation of a strict terminological matching of taxonomical ancestors. The first method was based on knowledge representation by considering semantic overlapping with an ontology against taxonomical ancestors that exists in other ontologies. The second method exploited the semantic links network with the structural similarities between ontologies as an indication of implicit semantics. These methods aim to discover semantic similarity by finding the equivalency between ancestors.

This approach introduced a novel measure to calculate semantic similarity between different ontologies. However, it did not consider descendants when calculating the semantic similarity between different ontologies. Doing so can help find more similarities between different concepts in multiple ontologies. Moreover, it will work side by side with the ancestors' similarities and new similarities as a result of considering the descendant's similarity.

Teng et al. [62] introduced a method called Semantic Overlap Ration of Annotation (SORA), which calculates the functional similarity of the GO context. This calculation uses three steps. First, SORA uses semantic specifications and coverage to compute the information content (IC) by deciding the IC's location but it is not based on a number of annotated proteins. Second, it exploits the combined inheritance and extended IC to measure the IC of any term. Finally, SORA generates functional similarity by exploiting the IC-overlapped ratios in terms. This approach employs the location in the GO to calculate the functional similarity of the gene product. They evaluate this method against five related methods and describe that they achieved the best results in comparison with these methods.

This method is a new way of calculating semantic functional similarity; however, it has a limitation: It is unable to exploit the information that is semantically useful, such as GO parents when calculating the IC between different terms. This can be exploited for several purposes such as increasing the similarity between different terms. The IC score of a term changes over time in a specific corpus and changes multiple corpora annotations. This method did not take into account such changes. In order to enhance the use of this method, an inference method should be employed to exploit semantic information included in the IC and use it to enhance the similarity. The second limitation can be addressed by including a tool that calculates the similarity between a pair of genes or terms due to their changeable nature.

Al-Mubaid and Nguyen [45] illustrated a new measure designed to measure the semantic similarity between biomedicine concepts by extracting the semantic similarity between a single ontology and multiple other ontologies. It is based on three basic principles: *the length of the cross-modified path between two concepts, a novel feature of widespread specificity of concepts in the ontology and the local granularity of ontology clusters.* In addition, this measure uses the Unified Medical Language System (UMLS) [63] as its framework. The methodology of such a measure depends on the length of the cross-modified path and shared specificity of concepts in the ontology; as such, these principles deal with elements of the ontology from different perspectives. The cross-modified path depends on both the length and depth of each element to determine the semantic similarity between other elements, so the authors considered the specificity of the ontologies' elements by exploiting each element's depth. They also used a "last common subsumer (LCS)" [45] to decide the novel feature of widespread specificity between pairs of elements in an ontology. In addition, they explained that the current semantic similarity measures do not use the local granularity elements enclosed in each concept. However, they studied the local specificity of a concept node to exploit any sub-tree or taxonomy tree contained in that concept.

There is a limitation in this approach which may obstruct the process of identifying semantic similarity. This approach is not fully automated; it depends on the user's decision in deciding the proper or primary ontology to be the base for calculating the semantic similarities between the selected ontology and the other participant ontologies. As a result of the disparity in opinions between humans, choosing the primary ontology will differ among users, which may decrease the accuracy of the semantic similarity result. Thus, adding a method to automatically decide the primary ontology will contribute to solving this problem and maintaining stability in accuracy when calculating the semantic similarity between different ontologies.

Alvarez Vega [64] introduced a new approach for calculating the semantic similarity between GO terms and gene products, and he dealt with proteins such as gene products. This approach represents an enhancement of the previous approach [65], which calculates semantic similarity between English words via the WordNet ontology, as the author adjusted the previous approach to deal with gene ontology and proteins. This method works by combining the pairwise semantic similarity between sets of annotated terms for two proteins or genes. Moreover, it takes a pair of proteins and designs a sub-graph that reflects the semantic relations between the given pairs through a comparison of the terms' annotations. Afterwards, the semantic similarity can be determined. In the final stage, all pairs that have semantic similarities are composed to decide the level of semantic similarity satisfied between each pair of proteins. Equation 2.1 shows the method of calculating the semantic similarity between a gene ontology term and proteins.

$$SSA(t_1, t_2) = \frac{sp_{sim}(t_1, t_2) + nca_{sim}(t_1, t_2) + ld_{sim}(t_1, t_2)}{3} \tag{2.1}$$

Equation 2.1: Calculation of Semantic Similarity (Alvarez [64], page 62)

Where sp_{sim} represents the shortest distance between two compared annotations, nca_{sim} represents the depth of the nearest ancestor and ld_{sim} is the level of similarity achieved between pairs. This method returns one of two values: 1 for high similarity and 0 for no similarity between the compared elements.

This approach introduced a useful method for calculating semantic similarities between different concepts within GO or between GO and other ontologies. However, this approach is not as accurate in calculating semantic similarity since it uses the pairwise methods, so it has the same problem as in [55]. Because, the pairwise methods employ the BMA method to annotate a gene, they have some limitations caused by the combined rules used in pairwise methods. Thus, in order to enhance this approach, a group-wise method [56] should be considered to calculate semantic similarities between a group of genes or proteins based on annotated terms.

Another method was proposed by Bollegala et al. [66] utilising a robust semantic similarity measure employing the Web to measure semantic similarity between entities. The main idea of this method is calculating semantic similarity based on page count with a lexico-syntactic pattern-text snippet returned by webpages. This method compared with several taxonomy methods for both edge-based and information-based content, and it provides strong results in measuring semantic similarity, especially in the taxonomy-based method. Moreover, it uses statistics provided by popular search engines to analyse texts on the retrieved content.

This method achieved interesting results in measuring semantic similarity with taxonomy-based information, which is not complicated and has standardised relations. Nonetheless, it has a limitation: It uses famous search-engine measurements to analyse specific text, such as queries. This is considered time consuming for calculating semantic similarity and thus it is not suitable for real-time applications. To overcome this limitation, this approach should have a method responsible for analysing short texts instead of using search-engine statistics for this task.

Nagar and Al-Mubaid [67] proposed a novel hybrid method that calculates the semantic similarity between GO terms. This method employs two main things namely, the structural Information Content (IC) of the Last Common Ancestor (LCA) and the Path Length Dependency (PL). This was done by firstly determining the root of the ontology, since GO has a DAG structure ontology; in other words, it does not have a root. So, they supposed the term or concept with the highest number of offspring is the root concept for this ontology, and the concept or term that does not have any offspring is a leaf in this ontology. Then, the authors computed the $IC_{structural}$ of the LCA based on this hypothesis. After that, they calculated the PL between the pair of concepts. Finally, they crossed the two scores to determine the semantic similarity score between the pair of concepts. The authors used pairwise methods to calculate the semantic similarity between pairs of gene or terms, by employing the best match average to find the maximum value of each row and column, and compute their average.

Even though this method proposed a new way of calculating the semantic similarity between different genes in the GO, it suffers from some limitations that affect its accuracy in calculating semantic similarity. It considers concepts with the highest number of offspring as roots for the GO, which is not quite accurate, since some concepts in the middle of the GO graph or ontology can have the highest number of offspring while still having one parent or more. This will cause some errors when calculating the shortest path between the GO terms, which is usually considered the root to calculate the shortest path between terms. Also, this method has the same problem as in [55] in that it is not accurate in calculating the semantic similarity, since it uses pair-wise methods. This method employs the BMA method to find the best match term, but BMA has some problems in the combined rules, as discussed previously. Thus, in order to overcome all aforementioned problems, the authors need to add a new concept and call it for example, "Thing". This concept should be the parent for any orphan concept in the GO and can be considered as the new root of the GO ontology. Also, the authors should use a group-wise method [56] to calculate the semantic similarity between groups of genes and overcome any problems that could be caused as a result of using the pairwise method.

Zhang and Haglin [68] provided a study that measures the capability of ontological scaling in analysing the semantic similarity between biomedical ontologies. So, measuring the semantic similarity in biomedical ontologies represents a potential role in understanding the depth of set of functions of gene or protein. Moreover, it supports in the systemic analysis of gene and protein data. This study was done by following several steps which can be summarised as i) calculating the semantic similarity between each couple of biomedical ontologies by applying well-known measures (i.e. Resnik, Jiang-Conrath, Lin, and SimRel). Then, ii) they applied the pairwise methods (MAX, AVG and BMA) in one ontology and with all concepts in the other ontology and then do the same with the second ontology. iii) They increase the scaling by 1 (increase the level 0 to be in level 1). iv) Then calculate the semantic similarity with the four different measures and applying the pairwise with the three metrics again. After that repeat all four steps and measure how much changing happened in the semantic similarity score. Finally, these steps will lead to conclude that selecting the appropriate scaling levels and similarity measures will reduce the size of ontologies without losing essential details in measuring the semantic similarity between the selected GO slims.

This approach has been applied to examine several semantic similarity measures and assess the semantic similarity between different biomedical ontologies. Also, the authors applied pairwise methods in different metrics i.e. BMA, AVG and MAX. However, the approach suffers from the same problem as in [55] since pairwise methods (BMA, AVG, and MAX) have some problems, as discussed previously in this section. Thus, this work should take into account group-wise method to overcome the limitations that may result from applying pairwise methods.

Furthermore, there are several other semantic similarity measures that have been designed for different purposes. For example, Li et al. [54] proposed a measure for semantic similarity which is a collection of non-linear information sources. Also, Thiagarajan et al. [69] proposed a semantic similarity approach using a spread activation network with matching concepts and multiple activation mechanisms.

2.3 Reasoning

There exist numerous resources, such as ontologies, corpora, databases and documents, leading to the proliferation of information on the Web. There are also inconsistencies between these different resources, which occur as a result of their various structures. Such inconsistencies necessitate reasoning techniques in order to extract valuable relations such as semantic relations and hidden associations which are included within these resources and can be used for different purposes. Exploiting these characteristics (i.e. semantic relations and semantic similarities) between different terminologies in different resources by applying some reasoning techniques on the extracted information will contribute to eliminating several obstacles in order to discover new facts and relations that appear as a result of this reasoning [70].

Furthermore, there are several reasoners designed to reason through ontologies, and can contribute to addressing inconsistencies. For instance, Shearer et al. [11] introduced one of the most popular reasoners: HermiT. This is an OWL reasoner which was developed based on a novel “hypertableau” calculus. This theory tries to reduce the number of considered models during the reasoning process. HermiT can handle sources of complexity (i.e. the number of constructed models and models that have been constructed by tableau, which are usually large), and it can reason through some ontologies within their descriptive graphs.

Although HermiT represents one of the most popular reasoners that has been used in several approaches such as in [71], it has some limitations in handling some query answers, such as checking conjunctive query answering. This is because of the limited computations for the query's answer that can be handled by this reasoner. So whenever the reasoner reaches a specific number of answers, it will not provide answers for the remaining queries. Moreover, this reasoner does not support the SPARQL[1] query language [72] or Jena[2] [73]. This represents a limitation for our approach and we have used the Jena framework and the SPARQL query to reason through different OWL files that we will process. To avoid such limitations, we have used the Pellet reasoner [10] which was included in the Jena framework and supports SPARQL queries.

Furthermore, reasoning has become required and been developed for different benefits. For instance, Okoye et al. [74] proposed a semantic rule-based approach that detects user interactions with the knowledge base. Then, it tailors a response based on adaptive rules stored in the user profiles. This work sets up semantic rules and descriptive logics queries to construct an ontology pattern that is capable of automatically computing many interactions that may happen in the knowledge base to test the regularity of the objects or data types. This work used reasoning methods to discover and infer through a learning knowledge base, which contributed to discovering new models or behaviours. Moreover, Torres et al. [75] introduced a method that performs qualitative reasoning in geographical representations. The reasoning was based on previous knowledge that was obviously modelled by using an ontology of an application. This method is called RAIN and concentrates on the conceptuality of concepts and relations that are included in the task domain. It suggests answers to questions for which a range belongs to a group of semantic descriptions and the level of the concepts relevancy to that domain. This method consists of two phases, namely, i) *Analysis and conceptualization*, when the prior knowledge-based reasoning demands are defined; and ii) *Inference*, when a group of ordered domains is detected and considering the proximity or similarity of the input descriptions.

Jamalabadi et al. [76] illustrated a reasoning technique for fuzzy classifiers as competitive interactive reasoning (CIR), which uses aggregated data supplied by all of the fuzzy rules and controls decision limits as if membership functions

[1] `http://www.w3.org/TR/rdf-sparql-query/query`

[2] http://jena.apache.org/documentation/inference/

have been directly adjusted. This technique is mathematically calculated by linear transformation and collects competitive interactions gained from brain neuronal columns. Moreover, this technique supports the idea that CIR facilitates the formulation of the fuzzy rules and combining the expert knowledge by impounding the devastation effects of noisy rules or expert knowledge. In the following sections, we will discuss different aspects of reasoning through various resources.

2.3.1 Reasoning with Multiple Resources

Reasoning with multiple resources (e.g. ontologies and corpora) is an increasing need due to the large number of resources available on the Web. These resources are rich with semantic relations and hidden associations that need to be extracted from diverse resources, such as ontologies, and used for different purposes such as providing recommendation services (i.e. recommendation on articles). Extracting such information and applying a reasoning process on them will help retrieve more accurate results and recommendations. However, there are several challenges that may obstruct the performance of reasoning through different resources. These resources may have different structures and may not necessarily be compatible with each other, which may lead to inconsistencies during the reasoning process [77]. This section will discuss some of these approaches in order to provide an overview of the methods that have attempted to address such challenges. It also will describe the limitations of these proposed solutions.

Zhong et al. [78] illustrated the event-ontology reasoning method, which has been implemented based on the event-influence factor. This method exploits the event class and event model to reflect some inference relations as a result of introducing a new event. This method is able to perform event reasoning from multiple-level strategies. Moreover, this method can implement reasoning in associated lengths between event classes. This reasoning leads to the discovery of an event that will happen after a series of events occurs as well as the event-class relations and different elements of the event classes. So, this method infers some results or events that occur as a consequence of a series of events that come from either single or multiple events. This method was evaluated by selecting five domain events from different ontologies; it then applied reasoning cases to these five domain events. The top five answers were references or inferred results.

This method has a limitation: the inability to control the number and type of inference results, which may consume time and effort as well as machine memory when applied to a large number of events. This method does not apply anything able to control the number of retrieved or type of inferred results. For example, considering semantic rules able to determine the type of relation to be inferred or applying something to determine the number and type of retrieved results (such as a SPARQL query). This limitation can be overcome by considering semantic rules or a SPARQL query, which will contribute to providing better results.

Fang et al. [79] introduced a method for performing contrastive answers for inconsistent ontologies. This method is an answering system from multiple, inconsistent ontologies that uses contrastive reasoning. It answers each query provided by the user, but each answer has an extension that represents a surprising answer linked with the original. All these parts are called contradictive answers, which comprise an original formula (or the significant answer for the given query), a contradictive formula (or the contrast with the original formula) and a clarification formula, which represents a clarification of the contradictive formula and sometimes is ignored or deleted when the contradictive formula is clear and does not need further clarification. This method has been implemented in the Large Knowledge Collider (LarKC) framework and Contrastive Reasoning with Inconsistent Ontologies (CRION) as a plug-in reasoning component.

This work has a limitation: It was designed to handle simple answers related to text only, which has no semantic information to be exploited. Therefore, this method is not sufficient for ontologies containing complex or nested relations. The method also cannot handle ontologies with contradictive links, such as *however* or *but*, as the reasoning of this method is designed for simple text only. To enhance the utilisation of this method and make it able to deal with different types of ontologies with contradicted conjunctions, the semantic relations should be considered when answering user questions.

Serafini and Tamilin [80] proposed a distributed reasoning technique for multiple ontologies called “Distributed Reasoning Architecture for a Galaxy of Ontologies (DRAGO)”, which is a combination of semantic mapping for local chunks with single ontologies. This technique follows a peer-to-peer architecture where each peer contains a set of ontologies and mapping. For privacy and security reasons, the reasoning was performed in a special area and a bridge of rules was published as a result of the reasoning, which can only be called through local reasoning service due to privacy reasons. The DRAGO system contains several peers called DRAGO Reasoning Peers (DRP), where each peer consists of components designed for a specific task. Namely, the ***Registration Service*** is an interface for creating, deleting and updating registration for ontologies to be assigned with local chunks, the ***Reasoning Service*** allows dealing with registered ontologies, the ***Registration Storage*** holds access information for all registered ontologies and the ***Registration Manager*** receives registration orders and then checks whether the URI is registered with DRP or not; if it is registered, then it would execute the reasoning process.

This technique suffers from some limitations which affect its utilisation. It is based on a manually created ontology, where any mistake in this ontology may make the reasoning process inaccurate. Moreover, the bridge rules designed to deal with local services contain some constraints related to privacy and security. These constraints cause difficulties in data assigned between concepts in different ontologies. This issue can be addressed by reducing the number of constraints in the bridge rules to perform data assigning easily and effectively. Local reasoning is an effective idea in terms of security and precise results; however, any edits in external concepts or relations will cause the local reasoning service to be insufficient or inconsistent and requiring recompilation. To avoid such problems, their method needs to have a modular ontology that will be able to examine the reliability of the ontology due to any changes.

Another work that discussed reasoning that included multiple resources, which were ontologies for task and training events from the military, was introduced in [81]. These resources suffered from several problems, some external (e.g. different message format between resources) and others internal (e.g. different concepts, understanding with positions or entities). This work used ontologies to describe its resources and introduced an automated approach for military training events. It has two main goals. First, it focusses on expanding the representation of the hierarchical tasks of OWL; these tasks are based on two concepts: composition and refinement. Second, it provided a description for military events by introducing automated reasoning to discover and handle problems, such as potential interoperability between heterogeneous systems, with appropriate solutions such as the Semantic Web Rule Language (SWRL) [82] rules, which was able to find out all individuals that may represent a connection point between heterogeneous resources. Elenius et al. [81] discovered some drawbacks in unary and binary predicates (i.e. predicates are OWL classes or properties) as a result of applying SWRL in their work. The limitations are the lack of the ability in producing novel individuals after evaluating rules and the lack of a failure diagnosis. As a result of these problems, Elenius et al. [81] proposed a solution to overcome these limitations called *allKnown*, which returns all known values of different individuals and eliminates the mentioned problems.

This method, provides substantial solutions by introducing the *allKnown* tool, which returns all known values of many individuals between different resources. This contributed to solving several problems such as interoperability between different systems. However, the provided solution fell short due to the inability of SWRL to provide rules that cover all available known values or create predicates, especially with the complex rules. The drawback of SWRL in the mentioned point complicates dealing with some predicates and causes some predicates to remain unaddressed. This problem can be solved by performing some alterations to *allKnown* and adding built-in SWRL rules. This will allow it to be able to address all predicates with specific rules because the built-in (which can be customised by the developer to handle all former problems) will define a set of RDF in the form of OWL arguments to SWRL to support it with an exclusive antecedent. Also, this problem can be solved by replacing SWRL rules with custom-built in Jena rules which are able to address all predicates in complex rules.

Radev and McKeown [83] illustrated a methodology that summarises news events. This methodology constructed a system called SUMMONS that is used by the Defense Advanced Research Projects Agency (DARPA) in the United States. It produces a summary for their events and extracts similarities, differences and many other documents related to specific events. Furthermore, SUMMONS contains several features such as providing a tool to extract brief, interesting topics for each user, publishing information in a consistent manner and supporting the provided information with data gained from online resources.

Behind the several services provided by SUMMONS, there are limitations, including that SUMMONS that does not generate summaries for completed sentences. Additionally, it was not able to generate summarisation for noun phrases. To eliminate such problems, new semantic rules should be considered, which should allow SUMMONS to provide summarisation for single noun phrases; this would enhance its functionality in dealing with the processed documents.

Pereira Detro et al. [84], illustrated an approach for enhancing semantic interoperability in health care through semantic enrichment of the event logs with the domain ontologies and by using Formalise Concept Analysis (FCA) approach. This approach follows seven steps to satisfy its intended target, namely, i) collecting an event log which contains information about the process that will be used for execution, and ii) determining the process-mining techniques that will be used to discover the process model. Then, iii) the Process Mining Framework (ProM) will be considered to automatically determine the ontologies associated with the elements in the event log. After that, iv) the produced ontology will be developed based on experts' expertise. Then, v) FCA will be applied to make the conceptual knowledge, which will contribute to a better understanding of interoperability between processes to give more opportunities to discover knowledge gaps. Moreover, this step is required for data incorporation, either manually or by using a semi-automated method for merging ontologies. After that, vi) FCA will provide a lattice that will be converted into a type of concept hierarchy by eliminating the bottom concept and introducing the ontological concepts for each concept and sub-concept under each element. Finally, vii) the resulting ontology will have an increasing knowledge, which will contribute to enhancing the semantic interoperability.

Although this approach provides an important attempt in enhancing the reasoning process (semantic interoperability) in the healthcare, it is not a fully automated approach. Since, it requests some developer or expert intervention, which may hinder its performance, especially in the fifth step, which requires intervention from the developer for data incorporation. This will be done manually or semi-automated by using a method for merging ontologies. Thus, in order to enhance the accuracy of this approach, this step should be done automatically by using a fully automated method to perform such tasks without any intervention or help from the developer.

Rakib et al. [85], introduced a framework for knowledge representation using OWL, 2 RL and SWRL rules to reason over multiple and heterogeneous resources. They submitted a set of queries to ensure that queries are compatible with the designed rules, and used the Pellet reasoner to make sure that all queries fit with the designed rules. The main steps for this framework to reach to its target can be summarised as: i) ontology specification, ii) collecting resources and conceptualisation, iii) model execution, and finally, iv) model evaluation.

This framework provides a substantial framework for reasoning over multiple heterogeneous resources. However, it has one main problem: the use of SWRL rules, which are limited in providing rules covering all available known values or create predicates, especially with complex rules. In order to overcome this limitation, a built-in rule should be defined with a method (i.e. designed in a high-level programming language such as Java) to assess the performance of the designed rule and to support complex rules. Moreover, another solution could be to replace the SWRL rules with custom built-in Jena rules, which will be more likely to handle all predicates in the complex rules.

There also are several other approaches concerned with multiple-resources reasoning; these approaches were designed from different perspectives. For instance, Serafini and Tamilin [86] developed a method for distributed reasoning through multiple ontologies, that were connected through semantic mapping; and also developed a reasoning peer-to-peer architecture algorithm. Kim et al. [87] provided reasoning for learning through the reading method, while Rangel et al. [88] illustrated reasoning for multiple-logic frameworks. Lu et al. [89] performed reasoning for multiple-description logics and user-defined rules, whereas Bouché [90] introduced reasoning for different points of views for business advice. Kirayama and Tomiyama [91] provided an integration object design over multiple ontologies and Kaneiwa and Mizoguchi [92] illustrated a framework for order-stored logic programming for multiple knowledge bases.

2.3.2 Reasoning with Multiple Bioinformatics Resources

Several bioinformatics resources (e.g. ontologies, taxonomies and corpora) have become available and may contain various information that can be exploited for different purposes. Reasoning between these resources is needed to discover some new information that may exist. However, this is not easy, as these resources may contain different types of relations and structures that may be incompatible. Thus, reasoning between different bioinformatics resources is needed to overcome potential incompatibility problems. This section will discuss some relevant works that have developed reasoning methods in particular for bioinformatics or biological data.

Chen et al. [93] created a reasoning framework that analyses relationships included in biological entities. They used an ontology for traditional Chinese medicine (TCM) and western medicine (WM) to construct a conceptual model for a biological network. Moreover, they created a data model that contains corresponding biological data integrated into a biological knowledge network. A reasoning method was employed to infer the potential biological associations between biological entities from the biological network, and this method exploits both the conceptual model and data model for such purposes. This approach constructed its reasoning method based on the MapReduce algorithm [94], which is a parallel programming model for big data processing and clustering. This method follows three main steps to perform reasoning between TCM and WM. First, it creates a unified ontology with large biological conceptual data. Second, for data integration, the data model is formulated as a large, linked biological network. Finally, distributed reasoning finds the associations between different biological entities. Performing these steps will allow discovery of associations between TCM and WM entities.

Although this framework helps to perform a reasoning process between large biological data, it uses the MapReduce algorithm, which has a limitation that decreases the efficiency of this framework. MapReduce has a one-way scalability design that is not sufficient for processing small data with complicated relations. This design is not flexible for more computational processes with complicated relations. To overcome these drawbacks, they should replace this algorithm with an alternative algorithm such as MapMatch [95] that can address these problems.

Tari et al. [96] introduced a method that discovers new drugs through automated reasoning. It exploits knowledge and facts to discover new drugs gained from literature and knowledge bases to perform this process. This reasoning is performed by scrambling molecular effects that are the results from drug-target interaction. It links with several diseases and drug mechanisms as domain knowledge in *AnsProlog*, which is a language suitable for automated reasoning. This method is based on three elements to perform automated reasoning and discover new drugs: *knowledge acquisition*, which is selecting a source and extracting facts that help to identify drug identifications gained from text mining; *knowledge representation*, which needs to gain logic facts from various resources and logic rules which represent drug mechanism properties; and *knowledge reasoning*, which links several sources and allocates order to the steps that lead to drug indications. Thus, this approach performs reasoning based on the action description and logic of fact, but this approach does not use a SPARQL query to perform the task.

This reasoning method has a limitation: It is not quite accurate in extracting specific drugs, in contrast with the SPARQL query, which allows the user to determine the exact properties of the required drug and discover only drugs related to user requirements. This helps the user save time and effort when discovering drugs. This limitation can be overcome by considering the SPARQL query to discover drugs instead of analysing a series of actions to reach a specific and *logical fact*.

Tsafnat and Coiera [97] proposed a method that performs reasoning over heterogeneous biological resources which are represented as different models. It examines the use of multi-models (i.e. impressive multi-scale computational models of biological phenomena) where these multi-models consist of sub-models called daughters. The method defined several “daughters models” that combined and swapped information to formulate multi-models. This reasoner for multi-models was constructed to overcome three different challenges: *model selection*, *composition* and *using the computer to construct the model.* In addition, such a reasoner can benefit from reasoning through diverse multi-models from different disciplines to reach a treatment or diagnosis for specific diseases. The method followed to exchange information between different daughters is done by providing middleware or an interpreter between the two daughters, thus allowing each part to understand the other and be able to interact. This produces a new model containing useful information extracted from both candidate models.

Nevertheless, this reasoning method has some functionalities that need enhancement. For instance, the composition stage should be extended to accommodate more multi-models, such as the social model (i.e. based on the relations between different concepts), which sometimes provide some clues for a new daughter model, which would be enriched by different knowledge from various perspectives.

Another approach introduced in [98] is concerned with interoperability between biomedical ontologies by expanding relations between different concepts in the biomedical ontologies and exploiting information in the top-level ontologies, which show a shareable fundamental for both classes and relations [99]. Moreover, they provided a method that performs automatic reasoning over biomedical ontologies and provides a definition for contradictory classes. It also aligns and formalises certain concepts of biomedical ontologies with fundamental classes at the upper-level ontology. This method is performed by employing biomedical ontologies in OWL format [8] for automatic reasoning, consistency authentication, and knowledge discovery. Also, it has a method that formalises biomedical ontologies by using OWL with upper-level ontologies, then it draws an ontology as a result of this process. Thus, the existence of an incompatible class in an ontology reflects a fault in either structure or class description.

Although, this approach has the aforementioned properties, however, it has some limitations. The main shortcoming of this approach appears to be at the stage relating the biomedical ontology to the relation in upper-level ontology. The relation assertion of the biomedical ontology is done manually. For example, assume that we have *has_part*, *has part* and *has-part* as biomedical relations. The approach of Hoehndorf et al. [98], requires someone to assert that all these relations are semantically equal. Hence, it would appear that the process of relating relations lacks automation; performing the assertion manually when dealing with very huge or multiple ontologies will consume significant time and effort. This problem could be overcome by creating a method that performs the assertion of relations automatically.

Samwald et al. [100] illustrated OWL ontologies and automatic reasoning methods that work side by side to assemble information that could help clinical pharmacogenomics and avoid several failure treatments or drug reactions. So, their methods were developed to achieve five targets: i) representing and formulating pharmacogenomics knowledge easily; ii) extracting all mistakes in definitions that could be found in pharmacogenomics knowledge; iii) providing an automatic diagnosis to patients' diseases; iv) connecting the patient to all pharmacogenomics instructions and decisions; and v) allowing their reasoning methods to discover any contradiction or overlap between pharmacogenomics and other resources.

Even though these methods contributed to addressing problems and eliminating several difficulties in order to avoid any failure treatments or drug reactions, this approach did not provide fully automated reasoning. Since, it was assigning top-classes to Protégé to perform reasoning on the assigned classes manually, which required developer intervention. Moreover, this approach is not generic enough to be applicable in other areas because it tailored ontologies based on pharmacogenomics knowledge, which does not allow this method to be applicable to other areas or domains. To overcome these problems, assigning top-classes and lower classes should be done automatically. Also, this approach should not build new ontologies for pharmacogenomics knowledge. It should adopt or use them and reason through them without building any ontology in order to make their reasoning method flexible and interoperable with other approaches.

Mallona et al. [101], proposed a model that merges ontology-based semantic methods with a query knowledge-base for human Alu elements. The knowledge-base for human Alu elements exploits both Sequence (SO) and Gene (GO) ontologies, and is dedicated to finding all functional genetics data in the genomic content of the Alu. This model uses the OWL ontologies and applies the Semantic Web Rule Language, specifically the RuleML rule language, to reason through the different resources involved this model.

Even though this model provides a new way of retrieving data from different resources by using a set ontology-based semantic method to query the knowledge base for human Alu elements, this approach has a problem that may weaken its performance and accuracy in retrieving the required results. The RuleML language cannot handle mathematical operations, such as equal, greater than, etc. yet these represent an essential need, especially when comparing two concepts or elements. Moreover, this rule is not quite expressive with some rules that exist in complex domains. This weakness can be addressed by using a method designed in a high-level programming language such as Java to pre-process the data that will go through the RuleML rules. To overcome these limitations, another type of semantic rule should be considered, such as Jena rules, which do not suffer from such problems and are able to return better results.

Zhang et al. [102], introduced a hybrid method that aggregates machine learning algorithms with trigger words and syntactic patterns for mining drug enzyme interactions (DEIs) from biomedical literature. EDI relations are mined to reason and then infer all useful drug-drug interactions (DDIs), relying on drug-enzyme ontologies that consolidate biological knowledge. During the reasoning stage of this method, two main classes or concepts are defined in the DEI ontology: drug and enzyme. Then, any drug or enzyme found will be added as an individual under these classes respectively. The authors performed this ontology in OWL (Web Ontology Language) and then applied chain rules using the Hermit reasoner to infer new data that can be found as a result of reasoning through drug-enzyme ontology.

This method used a new technique combining machine learning algorithms with trigger words and syntactic patterns to extract EDIs from biomedical literature. Then, it used these extracted relations to infer new DDI relations through a reasoning process that will be applied over drug-enzyme ontology. However, this approach does not clarify what types of semantic rules are used, since some of the semantic rules have limitations such as SWRL, which has limitations that discussed in the previous section. Furthermore, this method also needs further enhancements to overcome other limitations. Firstly the limitation is caused by the Hermit reasoner, which is used to reason through drug enzyme ontology. The problem with this reasoner is that, as mentioned before, it is unable to handle some query answers such as conjunctive query answering, since only provides a limited number of computations for the query's answers. Therefore, replacing this reasoner with the Pellet reasoner contributes to addressing this problem, since the Pellet reasoner does not have this problem. Finally, this approach needs further enhancement by applying SPARQL query over the processed ontology and using the Pellet reasoner (because the Hermit reasoner does not support SPARQL queries) to reduce the time and effort for inferring unwanted or useless facts and information.

There are other works that discuss reasoning with multiple bioinformatics or biological resources that appear to be limited to specific purposes or datasets. For instance, Kohler et al. [103] discusses automatic reasoning and evaluation for logical definitions, while Baader et al. [104] present a reasoner with description logic $\mathcal{E}L^+$. Finally, Horvitz [105] presents key concepts of automated reasoning for biology and medicine information.

2.4 User Profiles

In addition to exploiting semantic relations between bioinformatics resources, user profiles represent an important source of information that can be used in the process of providing recommendations to users. User profiles can be constructed based on different methods. The data used to construct user profiles can be classified into two types: static and dynamic. Static includes the data that do not change very frequently such as name and age. Dynamic information represents user preferences that can be collected explicitly or implicitly [106]. Moreover, Zayani [107] presents a way to categorise the genres of data in the user profile into several types. **Basic information** contains users' personal information such as name and address. **Knowledge** has all the navigated webpages that have been accessed by the user. **Interests** are gained from a collection of keywords or expressions, and **feedback** is formulated from a users activities such as number of clicks or time spent on a web resource. **Preferences** include specific nodes, pages or links that could have a specific style or colour favoured by the user. There are two methods for collecting data, namely explicitly and implicitly, and these will be discussed below:

Explicit Collection: Explicit collection is a manner of data collection where the data need to be entered via user participation [108]. There are various methods for collecting user profiles' data explicitly. Collecting data explicitly requires a user registration process for some kinds of applications [109] or it can take place through a browsing session, filling out forms or questionnaires or ranking products or webpages [110]. Regarding the previous categorisation for the types of data included in user profiles, such as **Interests** and **Preferences**, these data can be acquired from the user via several methods. For instance, some systems or websites, such as Pandora.net[1], ask their users to select their interests and preferences directly by preparing forms that contain boxes to be selected by the user. They may be asked about favourite categories during the registration process and tick a box that updates the user's preferences. Moreover, there are some websites that gain user preferences and interests through rating products or purchases from websites such as recommendation websites. This method has both advantages and drawbacks. The major advantage of this method is the precision of the collected data because all data in the user profile have been filled out by the user himself/herself. This feature increases the level of credibility for all stored information. Chaudhuri and Tewari [111] provide an approach combining online query-level ranking via the Internet with user feedbacks collected explicitly from users. Moreover, Nwana and Chen [112] introduce a new tag preferences measurement that uses explicit collection of user tag preferences. Nevertheless, there are some disadvantages including the fact that users sometimes do not pay attention while filling out forms or answering questionnaires. Because of their unawareness of these tasks' importance in enhancing the precision of the provided recommendations. Also, some users fill out their profiles with incorrect or inaccurate data for personal purposes, which may also cause mistakes in the provided recommendations. Finally, user preferences change over time, and with explicit collection, the user will be burdened with updating their user profile periodically.

Implicit Collection: User preferences can be collected by observing user behaviour through browsing [113]. By tracking user-browsed webpages, an idea can be drawn about the user's preferred topics, articles or products. In addition, there are other types of data that can be collected implicitly from users such as through searched keywords [114]. Tracking users in this way makes it easy to gain general or specific ideas about their preferences and interests. This method avoids any inconvenience to users, unlike being asked to fill out forms or questionnaires, which takes time and effort. The popularity of such methods has been increasing, as many projects have constructed profiles based on this method. For instance, Wang et al. [115] provide recommendations about products through users pref-

[1] http://www.pandora.net/en-us/

erences gained by observing user-purchased items. Employing such a method in personalisation and recommendation may enhance the recommendations accuracy and precision. Implicit collection also helps researchers and users from different perspectives and disciplines become aware of all new information available on the Web. Another example of implicit collection provided in [116] is concentrating on combining collaborative filtering (CF) recommendations with implicit data and user behaviour to provide more accurate recommendations on scientific papers. Furthermore, Anh-Thu et al. [117] proposed a method for constructing an online shopping centre recommender system based on collecting user's feedbacks implicitly. Implicit collection is similar to the other methods in that it has both pros and cons. The pros for this method include not bothering the user with collecting preferences and interests; implicit collection systems are automated. Moreover, this method contains up-to-date preferences because it does not require the user to update preferences or interests. In contrast, there are some drawbacks to applying the implicit method for collecting user data. It infringes upon user privacy by installing tools that observe user behaviour and interactions and requires greater effort to perform and apply machine learning methods such as those described in [106] to acquire data without disturbing users by applying the process manually. Another drawback is the way of collecting user data. It is not quite accurate since it depends on the applied algorithm for collecting data and the machine learning algorithms differ in their accuracy in retrieving data and in their ways of representing the retrieved data.

2.4.1 Modelling User Profiles

A user profile can be defined as a group of data relevant to a specific user that represent the person's identity, preferences, interests, etc. [118]. These data can be static (such as age or sex) or dynamic (such as interests or preferences), as discussed earlier in the previous section. Moreover, Amato and Straccia [119] defined the user profile as a set of preferences for each single user. Also, Cornelis [120] illustrates that the user profile consists of data from which the user's preferences can be concluded. Finally, the Oxford Dictionary defines it as a set of data that represent user habits, preferences, interests, etc. particularly products or services

[121]. User profiles can be content user profiles or contextual user profiles; while the former is only concerned with the data, the latter is concerned with the user's data, time and location. However, this thesis will focus on content user profiles and will consider contextual user profiles in future work and enhancements. Furthermore, the items or preferences in the user profile are usually assigned with weights that reflect their importance and priority to the user. Users usually prefer items with high weights and do not prefer items with low weights. User profiles can be represented in different ways that represent four types, namely, *i) weighted keywords, ii) semantic network, iii) weighted concept (ontological representation) and iv) association rules.* These types will be described and discussed in more detail later, in section 2.4.1.1. User profiles represent a fundamental factor in recommendation and personalisation systems, which depend on user profiles to provide personalised services. Thus, without user profiles, these systems will not provide quite sufficient services. These systems will be discussed in more detail in section 2.4.1.2, which will show how these systems exploit user profiles to provide services (recommendations or retrieve results) to users.

User profiles follow three main stages that allow for successful exploitation of the included information. The essential stages for modelling the user profile are representing, exploiting and learning [122]. Researchers have developed different methods for each of these stages, and we will review the most relevant ones below along with their advantages and disadvantages. The following subsections illustrate the main stages of modelling the user profile.

2.4.1.1 Representing the User Profiles

There are many ways to represent the user profile such as weighted keywords, semantic network, weighted concepts or association rules [123]. So, we will discuss each one of them in more detail below:

- **Weighted keywords** is a popular method of representing profiles because these keywords can be obtained from Web documents. They represent the users' interests, and each interest has its own weight, which is represented numerically and reflects the level of importance of each interest to the user [123]. There are several systems that have considered such a method in representing their user profiles. For example, Salton and McGill [124] constructed an approach that adapted and expanded user profiles based on the weighted-keyword method, which has been used for creating a learning algorithm based on generic algorithms. The algorithm used was the *tf * idf* (Term Frequency- (Inverse Document Frequency)) [125] as a weighting schema for the keywords. Furthermore, Alaofi and Rumantir [126] introduced a personalised system that exploits the units in which students are enrolled to generate a weighted keywords profile for each student, which is used to estimate relevant resources from the library. One of the limitations of the weighted-keyword method is that it does not handle keywords of explanatory nature; this does not make it very suitable for user profiles.

- **A Semantic Network** consists of a set of nodes that are connected to each other to formulate a semantic network where each node in the network represents a concept. The semantic network appears to address the multiple-meaning problems for a single keyword as well as keyword ambiguity. For instance, the concept Java can refer to a programming language, but sometimes it refers a type of coffee, and sometimes refer to an island with this name; so the way used in a semantic network will eliminate such problems [123]. There are several approaches that apply this method in their user profiles, such as Gentili et al. [127], who researched an online filtering system for digital-library documents. Moreover, Lakiotaki [128] provided a method of classifying medical documents into documents for medical specialist and non-specialists, by formulating them as vectors and then employing Multiple Criteria Decision Analysis (MCDA) to exploit this data. This method uses semantic networks to classify and represent medical documents in user profiles. Although the use of a semantic network is effective in representing user profiles, it still represents network concepts following a static and simple method. As a result, some concepts which have conceptual meaning, such as health, which can be associated with different concepts (i.e. body, food, etc.), will still have the ambiguity. Because this method follows a simple and static method, and it will not be able to represent such concepts accurately.

- The **Weighted Concept (Ontological Representation)** method follows the same route of representation considered in the semantic-network method. However, this method is more expressive and dynamic than the semantic-network method. This is because each concept or node represents an abstract topic (i.e. provides some information about each concept as well as its relation with the others) that has a full description of each interest preferred by the user [123]. Several approaches employ this method. For instance, Skillen et al. [129] discussed constructing an ontological user profile for a context-aware, personalised approach for mobile environments. Reformat and Golmohammadi [130] illustrated a technique to update user profiles through browsing observation, wherein this approach considers an ontological user profile to represent their users profiles. Also, Yu et al. [131] provided a recommendation approach for enhancing learning services by exploiting ontologies to develop and represent a context-aware e-learning system. As such, this approach used ontologies to represent contextual information as well as user information (i.e. interests). Moreover, Luna et al. [132] provided an approach that represents the interactions between a user profile and its context for collaborative learning. This approach uses ontology to represent its user profiles.

- **Association Rules** were invented for a specific type of system concerned with patterns that need to be discovered. These rules can be represented by a group of webpages that associate or connect through a hyperlink [133]. For instance, Wong et al. [134] provided an approach that infers a user access pattern from logon history data mined by fuzzy association rules. Moreover, Wakita et al. [135] suggested another approach to recommend or select new brands of clothes that are relevant to user's preferred brand, by employing fashion-brand association rules. This method is not quite sufficient with a huge amount of data, since different densities of processed or received data can cause limitations in this method's reliability and precision.

2.4.1.2 Exploiting the User Profiles

This represents the applications that can be performed based on user profile modelling. These applications include *recommender and personalisation systems.* In this research, our focus will be on the former type of applications, and we will discuss some approaches concerned with the latter type of application.

Recommender systems can, in general, be divided into three types: content-based, collaborative filtering and hybrid systems [136]. First, content-based filtering extracts information from several resources that match or are related to its content [137]. For instance, Mooney and Roy [138] provided a book-recommendation system with content-based recommendations and machine-learning algorithms for text categorisation. Moreover, Alharthi and Inkpen [139] illustrated a content-based recommender system that recommends jokes to users, using WordNet synsets to enrich the recommender system with extra information that can enhance the accuracy of the provided recommendations. As a result, a content-based system needs a complete profile and is unable to make correct recommendations for new users who do not have complete profiles, which can be considered a limitation. Second, collaborative filtering is where the recommendations can be provided by exploiting the information overlapping between users' preferences and calculating similarity between the user's interests with other users' interests [140]. An example of this method was provided in [141]. They introduced a model called *filterbot* that used collaborative filtering for addressing the sparsity and early-rates problem by exploiting the tapping strength filtering techniques. Also, their model targets check spelling for different users. Al-Badarenah and Alsakran [142] suggested an approach that uses collaborative filtering to recommend courses to students taken by other similar students. This approach employs association rules and applies techniques to find patterns between courses. This model has some limitations because it is unable to predict relevant topics for any item without previous ratings for that item. Third is the hybrid, which represents a fusion of the aforementioned types (i.e. content-based and collaborative filtering). Freund et al. [143] classified the hybrid approach into seven methods of hybridization:

- *weighted*, where the score of recommendation can be measured based in all existing recommendation techniques in the system;
- *switching*, which uses standards to switch between recommending techniques;
- *mixed*, which is when different recommendations come from different recommender systems simultaneously;

- *feature combinations*, where different recommendation algorithms are pulled together into a single recommender algorithm;
- *cascade*, which needs a performed procedure, since each recommender system refines the recommendation by another recommender;
- *future augmentation*, which is when an algorithm uses a recommender's productions as inputs to produce recommendations;
- *meta-level*, which uses learning algorithms to formulate inputs for the other algorithms.

Melville et al. [144] provided a hybrid method combining content and collaboration. It used a content-based predictor for improving the quality of user data. Moreover, it employed a collaborative method for effective recommendations through personalised services. The limitation of this method is its inability to predict when it has complex data, because it uses the Naïve classifier, which is not accurate with complex data. Furthermore, Hao et al. [145] proposed a probabilistic-based hybrid recommender approach that uses both content and collaborative filtering to provide recommendations. This approach employs user ratings and items' topics that exist in the scope of product design firms to provide knowledge recommendations. This approach does not consider a threshold to determine the number of most similar users, which will cause a large number of similar users to be found. This may cause a weakness in the accuracy of the provided recommendations due to the abundance of similar users who have several preferred items. Thus, a threshold should be determined.

User profiles play a major role in personalisation systems. Basically, personalisation systems use information about the user that may exist in profiles or in other forms in order to provide tailor-made services to users. So, recommender systems that provide recommendations can be viewed as a type of personalisation system. Other personalisation systems can take the user preferences and then present websites or webpages according to the way the user likes, which may differ from user to user. Personalisation systems have been used for different types of applications that have different requirements ranging from electronic newspapers to web applications. Daoud et al. [114] illustrated an approach that uses long-term user profiles based on a collection of short-term sessions. They used such observations to provide personalised search services or re-rank users' preferences. They studied the way to learn long-term user interests by collecting the concept-based short-term that characterises related to the user search activities. Also, Vu et al. [146] introduced a personalised approach using Latent Dirichlet Allocation (LDA) for data extraction from documents and tested three time scales in their personalised system, namely *long-term*, *daily profile* and *session profile*. They employed these time scales in a re-rank system that returned results from a commercial web search engine. Their results show that temporal profiles can significantly enhance the returned results. Moreover, Micarelli et al. [147] categorised personalised searches into three types: i) *retrieval process*, which is concerned with developing a complete search engine, which can help in providing personalised results and can be distinguished by using an internal search engine; ii) *query modification process*, where a user's query is modified to enhance the retrieved results; and iii) *re-rank process*, where results are retrieved from a search engine then re-ranked according to the user's interests. Each of these types contains several applications.

2.4.1.3 Learning the User Profiles

This represents an essential stage in modelling user profiles. To learn user profiles, there are several methods that can be applied. These methods can be classified as machine-learning techniques, in which each method was designed for a specific type of data contained in the user profile. Pazzani and Billsus [137] reviewed some of the classified algorithms designed for learning user profiles through different data structures, including the Decision Tree and Rule Indication, Nearest Neighbour Method, Relevance Feedback and Rocchio's Algorithm, Linear Classifiers, and Probabilistic Methods and Naïve Bayes. Some of these algorithms are widely used in recommender systems such as the Naïve Bayes algorithm. This algorithm is probabilistic and based on the Bayesian theorem, which is efficient for high dimensionality input, and its functionality is to compute the probability between new content and the constructed dataset. Also, it is commonly used in text classification applications [137]. For instance, Swezey et al. [148] provided a system based on the Naïve Bayes algorithm that is able to push the latest up-to-date content for news articles in real time. Also, this system uses the agent to support recommendations on content, matching them with the web browser. Also, Blanco-Fernandez et al. [149] introduced a method based on the Naïve Bayes algorithm that infers new data and provides recommendations based on TV-content ontology. Moreover, Cui Cui [150] introduced a method for Chinese text classification using the ICTCLAS (Chinese lexical analysis system of the Chinese Academy of Sciences) to provide text segmentation and information cleaning. This method employs the Naïve Bayes algorithm to perform this classification. Although the Naïve Bayes algorithm has been used in several recommender systems, this algorithm is useful for simple data (i.e. not having completed and nested relations) and it is not quite accurate with data that are complicated in nature.

2.4.2 User Profile Adaptation in Recommender and Personalisation Systems

The adjustable nature of users' preferences and interests over time represents an important issue for recommender and personalisation approaches; this changeability makes it difficult to produce sufficient recommendations for users. Therefore, user profiles with static information will not be able to make accurate recommendations over long periods of time. There are other factors that can affect the precision and accuracy of recommendations such as browsing time or place. However, these factors will not be considered in this research, which only concentrates on adapting the profile (i.e. adding, updating and deleting). In the adding process, the user's preference is added into his/her profile and assigned a weight. Updating involves increasing or decreasing the preference's weight over time based on whether the user visits the website or not, according to a specific threshold determined by the developer. Finally, in the deleting process, a preference is deleted from the user profile when the user is no longer interested in a specific preference (this can be determined by the system developer, as some systems check the last time the user visited a specific website or the preference weight may decrease until it reaches the deleting threshold, which is also determined by the system developer). Adaptive user profiles will be discussed in the rest of this section.

Shahabi and Chen [151] provided an extended approach for the Yoda recommender system [152], where Yoda supports large-scale web-based applications with an online real-time recommendation system that is based on a hybrid approach. This extension allowed it to improve the accuracy of the provided recommendations. They based the system enhancement on different resources such as expert consultations, web navigation and user opinions. This approach is a hybrid system combining collaborative filtering and a content-based method as well as confidence values learned implicitly from user feedback. The approach is heavily based on the user profile and employs a genetic algorithm to utilise users' behaviour automatically to provide more accurate recommendations. As result of using genetic algorithm (GA) in this approach, there is an enhancement in the accuracy of the provided recommendations.

However, this approach is fully dependent on user profiles to provide recommendations. Some of these profiles contain nested or complicated data that cause unreliability in the provided recommendations.

Trajkova and Gauch [113], illustrated an approach that implicitly built an ontological user profile. So, the profile was observing a user's browsing routine to represent the user's preferences. They also focussed their efforts on increasing the approach's accuracy by maintaining the user's profile stability and identifying essential concepts easily and precisely.

Nevertheless, this approach is not fully adaptive, since it has a limitation. This approach concentrated on creating a user profile (i.e. adding items to the profile only), not adapting (i.e. adding, updating, deleting) the created user profile.

Webster [122] introduced the hybrid transitional approach called HyGen for ranking purposes. This approach was designed to extract associations between genes and diseases across three disciplines in order to discover new diseases. Moreover, it re-ranks discovered diseases unique to each user based on his/her preferences in the profiles. HyGen was constructed based on three elements: *Semantic Web*, *graph algorithm* and *user profile* to find associations between genes and diseases in order to discover diseases and provide them to each user according to the user's preferences. The graph algorithm was constructed based on pseudo-relevance feedback [153], which is an information-retrieval algorithm. This approach asks the user to define the "seed" that represents the core of the user's profiles, and then the user's profile information is explicitly and gradually collected to formulate the user's preferences.

Although the HyGen approach can help to discover diseases, it suffers from some weaknesses that decrease its ability to rank new diseases effectively. This approach is not fully automated, and it asks users at certain points to direct the behaviour of the system. For example, when determining the triple for a specific concept and when drawing the sub-graph for the user, it requires the user to determine the number of iterations or levels to draw on the requested graph. Moreover, it uses pseudo-relevant feedback to construct graphs, a method that has the drawback of requiring a longer retrieval process. Furthermore, this approach collects user information explicitly, burdening the user by asking him/her to complete forms; sometimes, users do not provide accurate information. In order to overcome these shortcomings, one solution could be to insert an automatic component in the sub-graph drawing stage, which does not wait for the user's decision to perform the drawing process. Such a component will enhance the approach's capabilities and fully automate it because it will not wait for or distract the system procedure by requiring user decision making.

Sheth et al. [154] illustrated a technique for diversifying recommendation results using a social network called “Social Diversity”. This technique exploits information shared between different users and user memberships in the social network within specific groups to provide diverse recommendations. Moreover, this technique gives users the opportunity to increase or decrease the diversity of their recommended results. It also allows users to receive diverse recommendations in specific searches such as recommended movies; in other words, the recommended movies will not be exclusive only to the genres preferred by the group but also will include films that are not very similar to some extent to those preferred by the user's groups.

Although this approach is different from the other approaches, its weakness is its inability to recommend users who are not members of any group in the social network and who did not rate any movie. Since, this approach was constructed based on a collaborative filtering method, which is insufficient for users who do not rate any movie and do not belong to any group. To eliminate this problem, this approach should be converted to a hybrid approach, which considers both content-based and collaborative filtering to provide recommendations. Thus, the user-browsed history can be exploited to provide recommendations.

Yoneya and Mamitsuka [155] introduced a new recommender system based on content-based filtering. Their work was focussed on providing recommendations for PubMed[1] articles. The PubMed article recommendation system (PURE) has an interface that allows users to interact with it and add/delete their preferred articles daily. Thus, each user needs to create a profile to be served by the PURE approach. This system clusters the articles provided by the users then sends daily e-mails about the new articles to them in the form of recommendations.

There is a limitation in this work, since this system is not able to handle the frequent changes in user's preferences, so recommendations will be sometimes provided based on old preferences. Because this approach is not able to provide recommendations for the user directly (i.e. while requesting the recommendation), where it sends them by email. This can decrease the accuracy of the provided articles because after a long time, the user may not be interested in the specific article topics. Moreover, this method is based on the added and deleted preferred articles which should be handled automatically because when the user is not feeding this method with the preferred articles, it will not be able to provide accurate results. To overcome the aforementioned problems, the user profile should be automated, and the articles should be clustered based on user preferences.

Middleton et al. [156], [157] and [158] developed a recommender system that provides recommendations for online academic papers. It uses a single source (i.e. ontology) to enrich a user profile and draw recommendations based on this enrichment. This system also considers user feedback to construct the user profile. Two recommender systems are constructed in this work: Quickstep and Foxtrot. Quickstep exploits ontological deduction to enhance the profile and uses an external ontology that improves user profiling. It considers a research paper topic based on the ontology of computer science classifications performed by a directory from the ODP. This system considered the k-Nearest Neighbour classifier for semantic annotations in the paper, associating them with the topic of the paper, including the ontology. The latter (Foxtrot) enhanced Quickstep by employing user feedback to enhance the recommender system. Moreover, it has an interface which allows

[1] http://www.ncbi.nlm.nih.gov/pubmed

users to interact with the system and e-mail notifications to keep the user aware of all updated papers.

Although this work is enhanced (by Foxtrot), it is still limited in its utilisation. This system does not take into account the availability of multiple sources of information (ontologies, taxonomies, etc.) to enrich the user profile. This may decrease the accuracy of the provided recommendations. Thus, this approach should be supported with multiple ontologies in order to enhance its performance to provide rich recommendations.

Mirizzi et al. [159] provided a recommender approach for movies that use Linked Open Data (LOD)[1] datasets, which represent an ontology for multiple resources. This method employs a Vector Space Model (VSM) [160] to deal with semantic information and has been used to support the recommender system. This recommender has been developed and connected with Facebook[2] as a plug-in to provide recommendations on movies, and it has been connected with user profiles to provide user recommendations based on his/her preferred movies. In this system, the user profile is created when the user adds a movie in the preferred list. So the user will receive recommendations on movies that are related to the preferred movies that exist in his/her profile. Furthermore, this system has been enhanced in [161]; they made a slight change in the method used to recommend movies. They divided the similarities of two movies by the sum of the $\alpha\rho$ (i.e. A weight which is given to each property which representing its value in the user profile) coefficient instead of dividing it by the number of selected properties, P. So, they claim that such a change will help to enhance the recommendation accuracy.

This method extracts some semantic information from the LOD and has been enhanced by another group of developers to make recommendations more accurate. However, it does not successfully employ the extracted information by performing further inferences to discover more triples, which can enhance the quality of the recommendations even further. Also, the user profile that is being created in this approach is not adaptable because it lacks the update and delete mechanisms. Thus, a user's profile will not be responsive to the frequent changes made by the

[1] http://linkeddata.org/

[2] https://www.facebook.com/

user. So, all the aforementioned drawbacks result in the provided recommendations lacking accuracy.

Ge et al. [162] proposed an approach that is ontology based for recommendations. This approach constructs a domain ontology drawn from heterogeneous resources and exploits it to provide recommendations. Users' long-term preference and interest ontologies are built based on user demographic characteristics and personal preferences. So, the similarity between the user's ontology and domain ontology is used to provide users with recommendations.

One main problem in providing personalised services in this approach is that the process of mapping user queries to the domain and user ontology is only done using matching and correcting the syntax of the query but not the semantic matching. So, this lack of semantic matching might cause classifying a query to a wrong concept, and hence the whole personalisation process might be inefficient. This method does not infer any semantic relations between concepts, but it uses a simple distance matching process. Moreover, this method does not have an adaptive user profile.

Shen et al. [163] illustrated a hybrid method to recommend webpages for users to improve the accuracy and diversity in a scientific knowledge-sharing platform. This method was content based initially to recommend webpages which calculates cosine similarities between different webpages, but it suffers from diversity, which leads to inaccurate results. Thus, the authors considered the collaborative-filtering method by using the Tanimoto coefficient for calculating similarity between webpages to work side by side with the content-based filtering to provide more accurate recommendations. So, they suggested that considering the hybrid method helped to enhance the quality of the provided recommendations.

This method ignored two main factors that can also contribute to enhancing the accuracy of the provided recommendation, namely, i) exploiting semantic relations or associations that may be found as a result of information overlapping between different webpages to enhance recommendations; and ii) constructing an adaptive ontological user profile that keeps the user profile updated and exploits knowledge that may be acquired from the ontology to provide more accurate recommendations.

Meymandpour and Davis [164] illustrated a hybrid recommender method for

MovieLens[1], which combines the collaborative method and semantic-analytics method for the LOD dataset in order to enhance the provided recommendations for users with most relevant movies. This approach uses LOD to calculate similarities between recommended items. They argued that this hybrid method will help to eliminate the cold-start problem when too few ratings have been found for specific users on the watched items.

Even though this method enhanced the quality of the recommendations, it still suffers from limitations which restrict its performance at the level of the accuracy in the provided recommendations. This method had exploited some semantic relations included in LOD, but in a limited way, as this dataset contains several types of relations (such as *has-part* or any other relations that could be discovered from information overlapping) that also can be exploited to enhance recommendations. Moreover, it does not have an automatic ontological user profile, where the ontological profile could help to infer new relations and information that could exist in the ontology combined with user preferences and the LOD dataset. This ontological user profile should work with an automatic method that is responsible for adding, deleting and updating user preferences which may change over time to keep recommended items updated.

Bianchini et al.[165] suggested a recommender system that generates food menus as recommendations. Their system exploited both a recipe dataset and semantic annotations to provide recommendations that were tailored to each user based on the preferences stored in the user profile. They expanded the *food.owl* ontology[2] for semantic definitions and classifications. This method is concerned with long-term preferences for constructing user profiles to determine preferred menus based on the user's previous selections of menus or recipes.

Although this system provides a modern and valuable method for recommending food menus. In addition, it exploits semantic annotations to enhance the accuracy of the provided menus and to tailor menus to the user based on his/her own preferences, it ignores an inference method that can go through different concepts and classes in *food ontology* to infer new dishes related to what the user used to eat or what is proper for the user's nutrition system. This is an important feature

[1] http://movielens.org

[2] http://krono.act.uji.es/Links/ontologies/food.owl/view.

that should exist or be exploited in any method that uses semantics, especially in recommendations. To enhance this system, an inference method should be integrated to it that can infer relevant foods or dishes in *food ontology* to the menus that the user used to order dishes from them. Moreover, it is only concerned with long-term preferences, without taking short-term preferences into account. This may lead to inaccurate recommendations when the user prefers to change his or her usual menus. To overcome this shortfall, short-term preferences should considered when constructing the user profile.

Erekhinskaya et al. [166] illustrated an approach that recommends the best publications to be read by a therapist after deep analysis of a patient's records and the papers previously read by the therapist. Then, the approach uses medical domain inference to infer semantically relevant profiles to recommend the best papers for the doctor to read that are semantically relevant to both the patient's case and the doctor's reading history. This approach takes the patient's records and publications as a list, and then finds the semantic matches between them to allow the doctor to read articles that are semantically similar to the patient's case. This work used a Natural Language Processing (NLP) tool to process the publications and perform a semantic index on them. Also, it used the Naïve Bayes classifier to classify lists of papers based on the therapist's needs.

This work provides a strong effort in term of recommendations provided to specialists to cover their needs. However, it has some limitations that disturb its performance. This work used the Naïve Bayes algorithm for classification, which is not quite sufficient for complex and nested data. Moreover, this work did not used an ontology (e.g. from the medical domain) to enrich the inference method with semantic information. This may lead to the discovery of new information to help obtain better results and more accurate recommendations. Furthermore, this approach does not involve constructing an ontological user profile. It can help and support the inference method with extra facts and information to provide accurate recommendations that match the patient's case and doctor's reading history. To overcome the former problems, the Naïve Bayes method should be replaced with a machine-learning algorithm that is sufficient for complex data. Moreover, a medical ontology should be integrated into the inference method and the user profile. It will lead to better and accurate results that are compatible with doctor's readings and the patient's medical situation.

There are several approaches which are concerned with recommendations or re-ranking, such as in [167], which introduced the collaborative-filtering recommender approach to capture user interactions with different tools and resources. So, they use different datasets (e.g. MovieLens, Book-Crossing[1], or EachMovie[2]) to provide an evaluation and comparison between different recommendation algorithms for technology-enhanced learning (TEL). Zhuhadar and Nasraoui [168] provided personalised search based on a user-centric recommendation engine, and one of the problems that their approach addressed was providing recommendations to users even when the system has not been used by the user before. This was done by creating an initial profile while the user logged into the system based on his/her personal information, such as department or teaching course. Hameed et al. [169] designed a recommender system that addresses the same problem (i.e. customising recommendations based on user preferences or interests).

2.4.3 Specialist Search and Recommendations

Search and recommendations are very similar and have the same target. Their target is to fulfil the user's requirements and provide him/her with required or useful information to help the user discover new facts and gain extra knowledge. Web searches started on 1994 by McBryan [170] and were then followed by several search engines, such as Yahoo and Google [171]. These engines tried to enhance the quality of web searches and the accuracy of the retrieved results. Moreover, recommender systems have provided useful information and results to users without submitting any queries or making any effort to find the required information. The first recommender system was suggested in [172] and used a collaborative filtering approach [173]. The idea behind this was to recommend items to users that users who are similar to the recommended user prefer.

After that, a new recommendations method appeared, representing a combination of search engines and recommender systems, in which the searches or keywords are exploited to enhance the accuracy of the provided recommendations.

[1] http://www.informatik.uni-freiburg.de/ cziegler/BX/

[2] http://grouplens.org/datasets/eachmovie/

For instance, Edmonds et al. [174] provided a news recommender system that allows users to search for information about specific topics by clustering the system to provide recommendations concentrated on that selected topic. For example, if the user is interested in sport, he/she will want to have recommendations on news that discusses something about sport. Thus, this system considers "sport" as a search or keyword given by the user and then clusters all recommendations on that selected word or key search. Hence, the user will not be recommended sources on any topic that discuss politics, weather, etc. unless these topics have a relationship or connection with sport. Furthermore, [175] suggested a movie and music recommender system that leverages the diversity between users' behaviour. So, it considers it as input data that the recommender system can exploit to provide more accurate and varied results. Moreover, this approach uses a method that adapts the variety of the search results to the recommender system through a refinement undertaken on the search query. Then, it determines the diversity between user profiles to produce recommendations based on an identical subset from the user preferences, which is called a sub-profile. These sub-profiles will be joined to generate the ultimate recommendations.

In contrast to the previous examples, Wang et al. [176] did the opposite by proposing a recommender system that used a conditional preference network (CP-net) to enhance the quality and accuracy of the search and provide personalised services to users. Moreover, systems such as Google API[1,2] provide both search and recommendations separately to support users with searches or recommendations, although they are still limited. Since, they do not exploit semantic information and relations to enhance either searches or recommendations. Therefore, all of the aforementioned works support the idea that both search engines and recommender systems have the same goal, as mentioned earlier in this section. Since, both of them try to provide useful and accurate personalised services that cover all users' needs to find required or useful information.

Furthermore, Kim et al. [177] suggested a semantic-based health recommender system that exploited personal health records (PHRs) to enhance the provided recommendations by exploiting both patient's PHRs and queries about the problem

[1] https://cloud.google.com/prediction/docs

[2] https://developers.google.com/custom-search/?hl=en

he/she suffers from. This approach employs ontologies to provide recommendations by showing a list of different ailments that are relevant to the patient's history and his/her submitted queries. This system also provides personalised summaries and videos about different ailments related to the patient's medical history. This is to avoid bothering him/her with manual searches in the system to find these summaries and videos. This system contains four main processes, namely i) the *Query and PHR Mapping Module*, which maps user queries with personal health condition (h_c), which are produced and revised by the data maintained in each patient's PHRs and web-based services; ii) the *Query & PHR Processing Module*, which determines the set of all personal expected ailments; iii) the *Summarising & Refining Module*, which is used to summarise health articles and refine videos based on meta-data; and iv) *Ranking Modules*, which are used to sort ailments into critical and non-critical, using the classification provided from the CDC in the U.S. for such purposes.

Although this system provided a useful recommender system that exploits patients' searches to enhance and exploit semantic information for recommendations, it does not apply SPARQL queries to retrieve data from their resources (ontologies). Retrieving all data can be useful for small resources, but it will be insufficient with huge data or big data, such as those in bioinformatics, and it will consume machine memory. Moreover, this approach lacks of a dynamic user profile which will cause the user to revive inaccurate recommendations that may not compatible with his/her needs. Thus, in order to enhance the performance of this approach, the authors should apply SPARQL to determine specific data to be retrieved from resources and avoid consuming the time and effort of their machines. Also, they should consider a method that provides dynamic user profile to provide up-to-date recommendations that fit with user needs.

Moreover, Livne et al [178] provided a content-based recommender system called *CiteSight*, which provides tailored, personalised citation recommendations to author groups for different assigned tasks. Thus, this approach supported the online tasks with hidden recommendations that were provided to the researchers directly during their tasks. Moreover, this approach treated the recommendations in the background to recommend them later to the offline tasks. This approach was an attempt to concentrate on enhancing time response and recommendation accuracy when providing this service. *CiteSight* uses graph-based or content-based recommendations to recommend authors with different citations included in the papers he/she cited before or other papers of authors who already included in the references of the current paper while the author writes his or her manuscript. It allows the user to provide a keyword about the paper he/she is interested in and then shows a list of all relevant recommendations (i.e. inline recommendations); whenever the user clicks on any paper in the recommendation list, the preferences will be updated based on this click. However, if he/she is not happy with the provided list, then he/she will need to add more text to enhance the accuracy of the provided list of recommendations. Moreover, there is a service, global recommendations, in this approach which simulates a principle of e-commerce, which supposes that if a user buys an item, then he/she will buy these items. Thus, this method supposes that when a user cites a paper, he/she will cite other papers that were cited by other users when those other uses cited the original paper.

The services provided by *CiteSight* help many researchers conduct research, write their papers and complete their publications. However, this approach concentrates only on syntactic matching and ignores semantic relations and associations that could be found among different papers. Exploiting semantics will enhance the accuracy of the provided recommendations, which is one of the desired goals for the *CiteSight* approach. Additionally, it will help the user discover more papers that may help him/her in his/her academic writing. Thus, to enhance this system, a method that extracts and exploits semantic similarities among papers should be added to further enhance the accuracy of the provided recommendations.

Saaya et al. [179] provided a development for a search method called *HeyStaks*, which represents their previous work in [180], which exploits user experience searches into *staks*. These *staks* are exploited in order to provide recommendations to the user while the user searches for sources. The recommendations of this approach added to the result of the search such as in Google, Yahoo, etc. based on the *staks* determined by the user and it will be classically shown in the retrieved results. Thus, users should select their *staks* during the search in order to retrieve recommendations concentrated on their queries and the selected *staks*. This work uses *HeyStaks* as an attempt to compile *staks* automatically instead of assigning the task to the user to complete manually. This was done by considering the vital *staks* based on search contexts (e.g. retrieved results and submitted queries). Authors assume such an enhancement contributes to improving the accuracy of the provided recommendations.

This approach has a limitation: it is unable to provide recommendations to the user unless the user submits a query. Suppose a user has a vital *stak*, but he/she does not have a specific word to search. In this case, this approach will not be useful, since its recommendations are conditionally connected with the user query. Thus, without submitting a query, this approach will not work. This system provides “re-ranking results” based on vital *staks* and user queries, but it does not provide recommendations. In order to overcome this limitation, researchers should consider a method that can provide recommendations whether the user submits a query or not. When the user submits a query, the recommendations should account for the submitted query and the vital *staks*, but when he/she does not submit anything, the recommendations should account for the vital *staks* only.

Table G.1 illustrates a comparison among sets of classical recommender systems, semantic recommender systems and exploiting-specialist-search recommender systems, equipped with different features which distinguish each from the others.

Finally, although the aforementioned works (in this section and in table G.1) sincerely attempted to enhance recommendations by exploiting semantics among resources, specialist searches or user preferences stored in the user profiles, they still have some limitations that weaken their performance. Thus, none of the discussed works reason through different resources in order to extract semantic relations (i.e. sibling or semantic similarity) and hidden associations which occur as a result of information overlapping among multiple resources, nor do they exploit the semantic relations and hidden associations to enhance the accuracy of the provided recommendations. Moreover, none of these works provided a recommender system that could support a specialist domain by employing researchers’ searches, preferences (which are up to date and represented based on ODP ontology) and the exploited semantic relations (sibling or semantic similarity) to provide more accurate recommendations. This exploitation for both semantic relations and associations may lead specialists to discover or read new articles that have not read before. Therefore, this thesis is exists to cover and address all gaps and limitations which exist in the discussed works.

2.5 Conclusions

To sum up, this chapter has discussed and examined several relevant works from many relevant areas, such as computing semantic similarity, mapping ontologies, reasoning through multiple bioinformatics resources, and user profile adaptation for recommendations and personalisation. Some of these works have covered similar aspects to our approach, such as reasoning and semantic similarity methods; however, they did not cover all of the details that will be covered in our approach. Moreover, some of the works are very similar in the features of their methods, but they still do not consider all of the functionalities and features that our approach will have. For instance, none of the discussed works (e.g. [159], [161], [164], [100], [98], [81], [85], [62] and [122]) provide a recommender system that supports a specialist domain by exploiting semantic relations and hidden associations from multiple resources in order to support specialists with accurate recommendations and extra information that could be gained from the information overlapping between these resources. Moreover, most of the discussed works suffered from some limitations or did not consider some features, and none of the discussed works provided a recommender approach that was supported by a semantic method that exploited semantic relations (e.g. siblings and semantic similarities). These relations occur as a result of information overlapping among different resources with different structures, such as ontologies or unstructured corpora, to enhance the accuracy of the provided recommendations in specific domains such as bioinformatics. Moreover, none of these works have used a recommender approach which was also equipped with an adaptive ontological user profile to exploit both the semantic information gained from ODP ontology and the semantic information and relations gained from the inferred semantic network (to represent the overlapping information and relations between different bioinformatics resources). Then, tailor to each user recommendations as a personalised service which fits with his/her preferences. Furthermore, these works lack of some useful features such as a user-friendly interface that allows the specialist to narrow down the recommendations to a specific interest or to concentrate recommendations on it. Thus, all of the aforementioned gaps and uncovered points will be addressed and covered in this thesis.

The next chapter will discuss the conceptual framework for specialist searches. This will include a discussion of the multiple resource structures and will show conceptually how different components in our recommender system will deal with each other. Moreover, the methodology that was used to construct an adaptive ontological user profile for the specific field of bioinformatics. Then, it will provide the evaluation methods that will be used to assess the level of enhancement that is satisfied by our recommender system.

Chapter 3

A Framework for Specialist Search

3.1 Introduction

This chapter introduces the framework that will be used in this research which includes the personalised recommendation method, that represents the implementation of the contributions of this research. This framework consists of different elements that were aggregated together to reach the research goal. Thus, this section discusses (i) the conceptual framework, the included components and the relations between different components of our recommender approach, such as different bioinformatics resources, the recommendation engine and the user profile. Moreover, it discusses the methods that these components used to provide accurate recommendations to specialists in the bioinformatics domain, based on their preferences and search activities. Also, it addresses the different resources and methods that were used to make these resources compatible with each other. Moreover, it illustrates methods that could be used to extract information from Wikipedia and examines steps that were considered to convert text data into a set of concepts that descend from the same origin. This conversion allows the semantic method to extract, exploit and reason through information taken from Wikipedia and infer some new relations and information that may help enhance the quality of the recommendations. Furthermore, it covers the automatic adaptive ontological user profile that was used in our approach, which represents one of our system components and its use within our system.

After that, it introduces (ii) the personalised recommendation method, which is a content-based method that provides accurate results to the user by using both the ontological user profile and the semantic method. Finally, it discusses (iii) the evaluation methods that were used to evaluate recommender systems in general and the metrics that will be used to assess our prototype system. These metrics will allow us to establish the level of enhancement that could be achieved when considering our main purpose. That is to support researchers in a specialist domain with accurate recommendations by exploiting semantic relations and hidden associations among different bioinformatics resources and through their search activities.

3.2 Framework Components

As mentioned, this work consists of different components that work cooperatively to support specialist users in bioinformatics with more accurate recommendations that are individually tailored to each user. Figure 3.1 shows different components of our recommender approach as well as the different resources that will be processed to achieve this research's main goal. This figure was divided into several stages, from 1 to 4, in which each stage represents a data-flow diagram for a specific component in our framework. For example, Stage 1 illustrates how different resources are accumulated into a single format (i.e. OWL) in order to be processed by our semantic method. This stage is connected to Stage 2, in which these resources will be processed by our semantic method and reasoned, and then all inferred relations and associations will be represented as semantic network to be exploited by the user profile component, which represents Stage 3 in this data-flow diagram. The user profile has been formulated based on the user's browsing sessions (as in Experiment 6.3) or given preferences (as in Experiment 6.4). These preferences will be matched with the most similar concepts in ODP ontology in order to exploit extra information that may exist in this ontology and create an ontological user profile. Moreover, the preferences in the user profile will be connected to the most similar concepts in the semantic network, and the user profile will exploit the semantic relation (sibling or semantic similarity) to enhance recommendations. Finally, Stages 4 consists of a user who is looking for the recommendations and the recommender system, which is in charge of exploiting semantic relations stored in the user profile (i.e. Stage 3). Then, the recommender system recommends articles from the BMC corpus based on the exploited relations and preferences stored in the user profile that are relevant to the submitted query. Furthermore, this recommender system also returns articles as search results from the BMC corpus; these articles are relevant to the user's query and the concept determined as result of exploiting relation from the semantic network. They are shown based on the preferences stored in the user profile.

The user or specialist looking for recommendations will not be aware of all four stages. He/she will be aware of the last stage (4) only, in which he/she will directly interact with the recommender system. Thus, let us assume that a specialist submits a query about gene called "GO_0008247". Since we are exploiting a sibling relation, our recommendation method will return a list of articles that discuss this gene as recommendations and articles on a protein called "PR_000025402", which represents the first sibling for the "GO_0008247" gene from the user preferences aspect. Furthermore, the search engine result will return articles that discuss topics relevant to the submitted query and to the most similar concept stored in his/her profile.

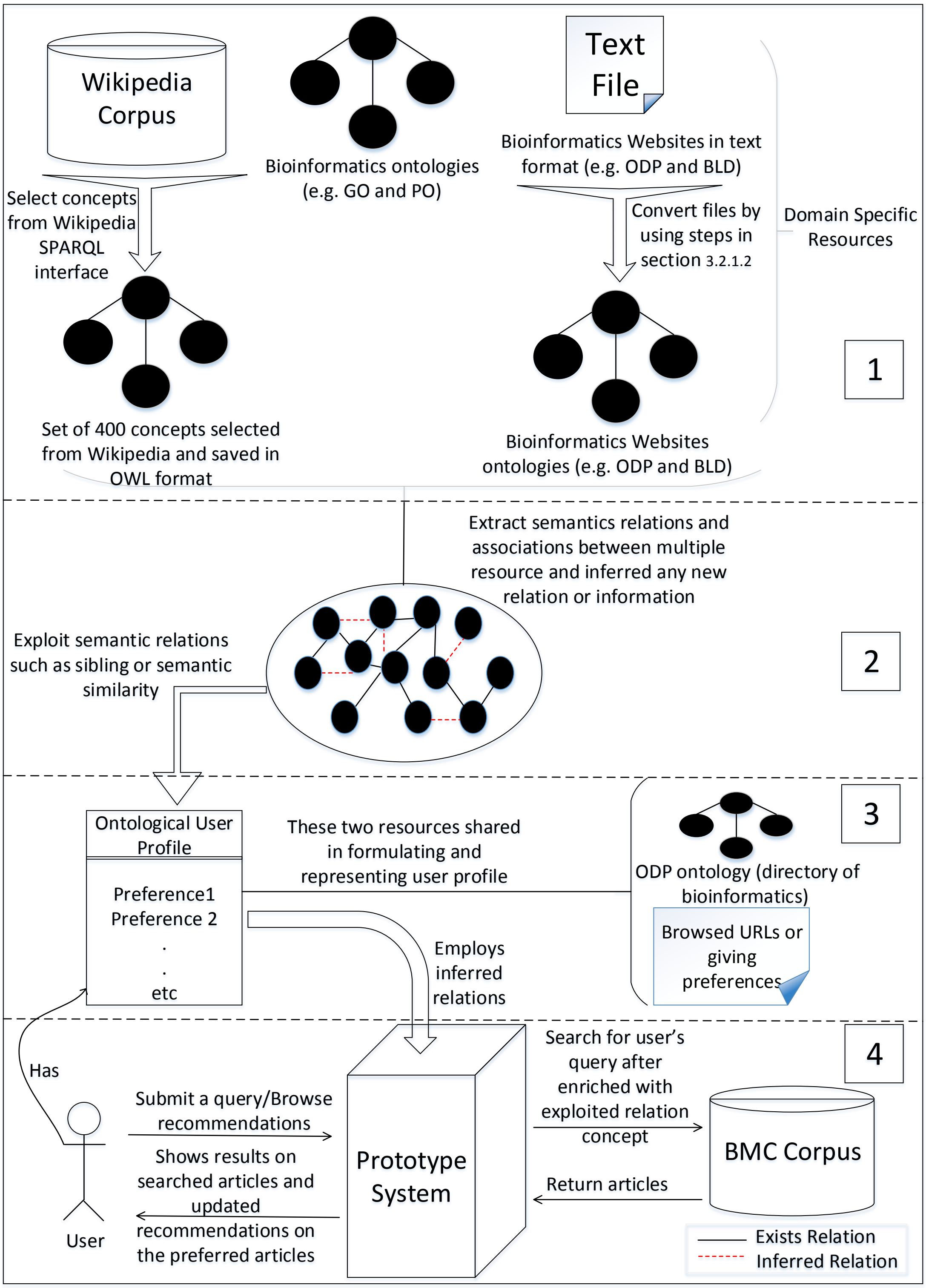

Wikipedia Corpus
Bioinformatics ontologies (e.g. GO and PO)
Text File
Bioinformatics Websites in text format (e.g. ODP and BLD)
Select concepts from Wikipedia SPARQL interface
Convert files by using steps in section 3.2.1.2
Domain Specific Resources
1
Set of 400 concepts selected from Wikipedia and saved in OWL format
Bioinformatics Websites ontologies (e.g. ODP and BLD)
Extract semantics relations and associations between multiple resource and inferred any new relation or information
Exploit semantic relations such as sibling or semantic similarity
2
Ontological User Profile
Preference1
Preference 2
.
.
etc
These two resources shared in formulating and representing user profile
3
ODP ontology (directory of bioinformatics)
Browsed URLs or giving preferences
Employs inferred relations
Has
Search for user's query after enriched with exploited relation concept
4
Submit a query/Browse recommendations
Shows results on searched articles and updated recommendations on the preferred articles
Prototype System
BMC Corpus
Return articles
User
Exists Relation
Inferred Relation

Figure 3.1: Data Flow Diagram for different Component of our Framework.

3.2.1 Processed Resources Structures and Preparation

We have domain resources which were selected manually by the developer based on the decided field. Thus, if the developer is interested in bioinformatics, then the selected resources should cover bioinformatics; likewise, if he/she is interested in computer science, then resources should cover computer science. These resources consist of set of files in OWL format, which will be processed by our reasoner to extract semantic relations and associations that may exist as a result of information overlapping between them. Each resource in this set represents a type of data; these resources will be described as follows:

- Website ontologies represent website directories that contain different concepts and categories in the field of bioinformatics, such as ODP and BLD. These resources appear in text format and need to be converted to OWL by following the conversion steps (which will be discussed in greater detail later in this section) in order to be reasoned and processed by our semantic method.

- Bioinformatics ontologies represent genes' GO and proteins' PO relations. These couples are in OWL and do not require conversion.

- Wikipedia's concepts (around 400 concepts from ODP, BLD, GO and PO were retrieved from Wikipedia and formulated as concepts in OWL format) represent a set of concepts selected randomly from websites and bioinformatics ontologies retrieved from Wikipedia. They were saved in OWL format to exploit the minor differences that can be found when different resources describe the same concept.

- The BMC, which is a bioinformatics corpus, was used to retrieve articles that the users searched for after enriching the user's query with concepts that represent the exploited relation. Moreover, a set of articles in this corpus that are relevant to the specialist's preferences will be used as a recommendation for the specialist.

As this work aims to improve recommendations in the field of bioinformatics and provide solutions that could enhance the quality and accuracy of the services provided for specialists interested in this domain. As mentioned previously in section 1.2, the bioinformatics resources used are PO, GO, ODP (branch of bioinformatics), BLD, Wikipedia and BMC corpus. Some of these resources went through the steps, that will be discussed later in section 3.2.1.2, to be ready for reasoning, such as ODP, BLD and Wikipedia, wherein these resources were not ready to be reasoned by our developed method.

3.2.1.1 Extracting Information from Wikipedia

Wikipedia represents an important source of information, since it has a large corpus that contains different articles and information about several concepts in different disciplines. Therefore, including this source in our research will enrich our resources with different types of information and relations that can be exploited in order to enhance the accuracy of the provided recommendations to the users. Moreover, Wikipedia is now the seventh most popular website and contains several textual pieces of information; however, there is a lack of approaches that exploit these data [181]. This is because Wikipedia has some limitations in its querying and search capabilities. Thus, it cannot retrieve a nested query that requests hidden results [181]. However, Wikipedia contains several structured information templates such as *"Infobox"*, which is a template located in the top left corner of the Wikipedia webpage [182]. It can be exploited in order to extract various types of information, such as semantic links between articles and other useful types of information.

There are many studies that focus on extracting information from Wikipedia such as [183], which concentrated on extracting the *Infobox* template and ignoring the other types of information. Völkel et al. [184] created a tool to be integrated with Wikipedia to extract the external links between articles in Wikipedia. There is an important extraction framework called DBpedia [185]. It can extract all structured information, such as templates, external links and all other information, and leave unstructured information without any further process. It also contains several pieces of linked data that help to extract different semantic relations. Moreover, it has been used by several applications that need to extract some data from Wikipedia [186]. For instance, Passant [187] introduced (dbrec) a music recommender system that was constructed on top of DBpedia that provides recommendations for around 39,000 bands and solo artists, and the linked data offered by DBpedia was exploited in order to build this recommender approach. Kobilarov et al. [188] illustrated the methods used by the BBC in integrating data and documents through BBC [1] domains and using the semantic web, and especially DBpedia; where DBpedia represents a controlled vocabulary and semantic support for the whole BBC.

The DBpedia extracts Wikipedia information and formulates it to be in RDF triples. Its extraction can be divided into four types, namely ***Mapping-Based Infobox Extraction***, which maps *Infoboxes* in Wikipedia into their terms in DBpedia ontology; ***Raw Infobox Extraction***, which maps *Infoboxes* in Wikipedia into RDF triples; and ***Feature Extraction***, which concentrates on extracting single features for Wikipedia articles such as labels [181].

Therefore, all of the aforementioned advantages of the DBpedia extraction framework motivated us to use this tool in our approach to extract information from Wikipedia. So, in order to use this framework, we needed a list of terms to extract information that can be found in the DBpedia framework. Therefore, a set has been prepared that consists of 400 terms. It has been selected randomly from our resources (ODP, BLD, GO and PO). Thus, the main goal of selecting these terms is to extract all information, such as *Infoboxes*, external links etc., from DBpedia as well as formulate an OWL file. So, the OWL file will contain exclusive information about our selected terms from Wikipedia. So, it can be exploited with our other resources (i.e. PO, GO, ODP, and BLD) in order to draw the semantic relations and hidden associations between different concepts contained in each resource. The process of extraction will retrieve several pieces of information about each term, and this information may share some details with other resources. Such sharing will contribute to adding extra features or properties to each concept, thus increasing the accuracy of the exploited data that will be used to construct the semantic network, and which will be enriched with valuable information about each term.

[1] http://www.bbc.co.uk/

3.2.1.2 Pre-processing and Dataset Conversion

This project deals with different types of resources such as corpora, ontologies and websites ontologies. These resources can be structured, unstructured or semi-structured in formats such as text, XML and OBO. Thus, our task was to convert these documents into a single format (i.e. OWL) to enable extraction of needed information, semantic relations and hidden associations. This process has been time and effort consuming due to the complicated relations in each resource as well as the huge amount of data in each resource. Therefore, conversion will allow for performing reasoning processes and exploiting semantic relations among different resources. In a real environment, there would need to be automatic wrappers that would be able to take information from one format and convert it into another, but for the purposes of this work, we had to perform these actions ourselves. The following steps were taken to pre-process and convert such documents:

- Analysing documents structure and determining any contained relation.
- Converting all assigned documents by designing a method that reads unstructured documents and converts them to XML format.
- Extracting relations contained in the documents and preparing the documents to be converted for RDF format to become meaningful documents.
- Producing documents in RDF format for each document and designing each schema to produce an OWL document that can be opened by different, current reasoners such as Protégé or RaserPro.
- Testing the produced OWL file and observing classes' relations, contents and visual representations in an ontology editor and reasoners, such as Protégé, ensuring that such a file is ready for any mapping, semantic similarity or matching processes.

The former steps are shown in figure 3.2 as follow:

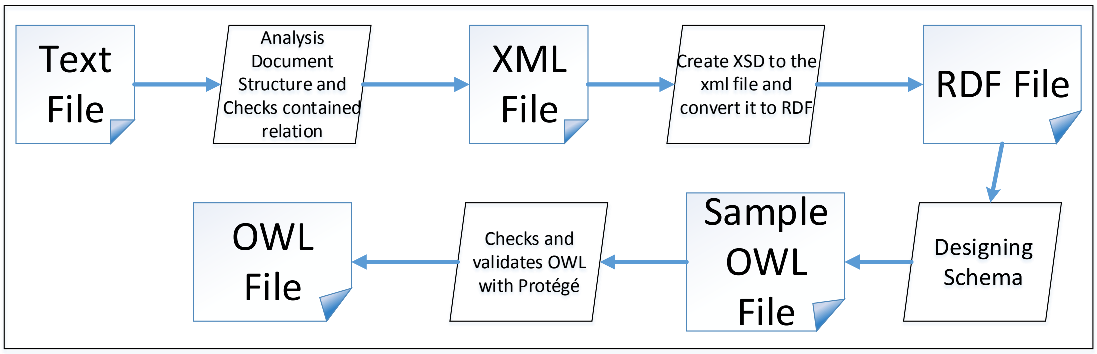

Text File
Analysis Document Structure and Checks contained relation
XML File
Create XSD to the xml file and convert it to RDF
RDF File
Designing Schema
Sample OWL File
Checks and validates OWL with Protégé
OWL File

Figure 3.2: Data Flow Diagram for Converting Text files into OWL files.

3.2.2 Prototype Recommender System

The prototype system is in charge of taking a specialist's query, extracting and exploiting relations to enrich the submitted query, then retrieving search results and updating the recommendations based on the uncovered and exploited relations and passing them on to the specialist. Figure 3.3 shows an overview of different components in our framework from a conceptual perspective.

For instance, our prototype recommender system will handle Tom's needs in example 1.1 as follows. When, he submits a query to search for his preferred concept, the system will enrich his query with concepts gained from the exploited relation from domain resources, then it will search for all articles that discuss his submitted query and enriched concepts. Moreover, recommendations will be shown as a set of articles that have the exploited semantic relations (sibling, semantic similarity, is_a,etc.) with Tom's preferences. Thus, in such a case he will notice all semantic relations between different concepts in GO and PO ontologies, such as the relation between PR:000027247 in the protein ontology and GO:0044445 in the gene ontology, but this information will not appear as a recommendation if our prototype recommender system is using just a single resource.

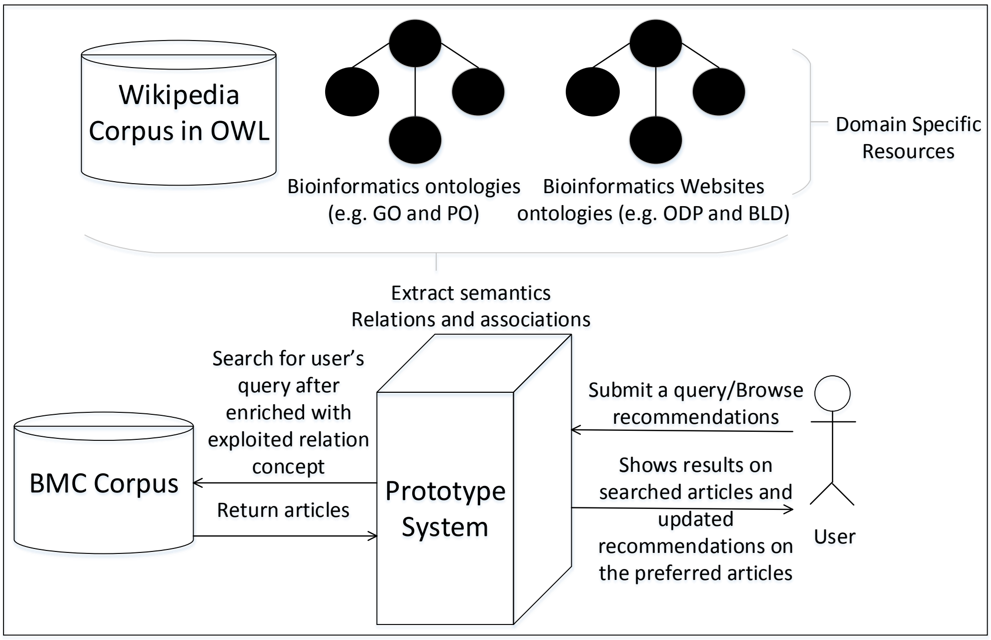
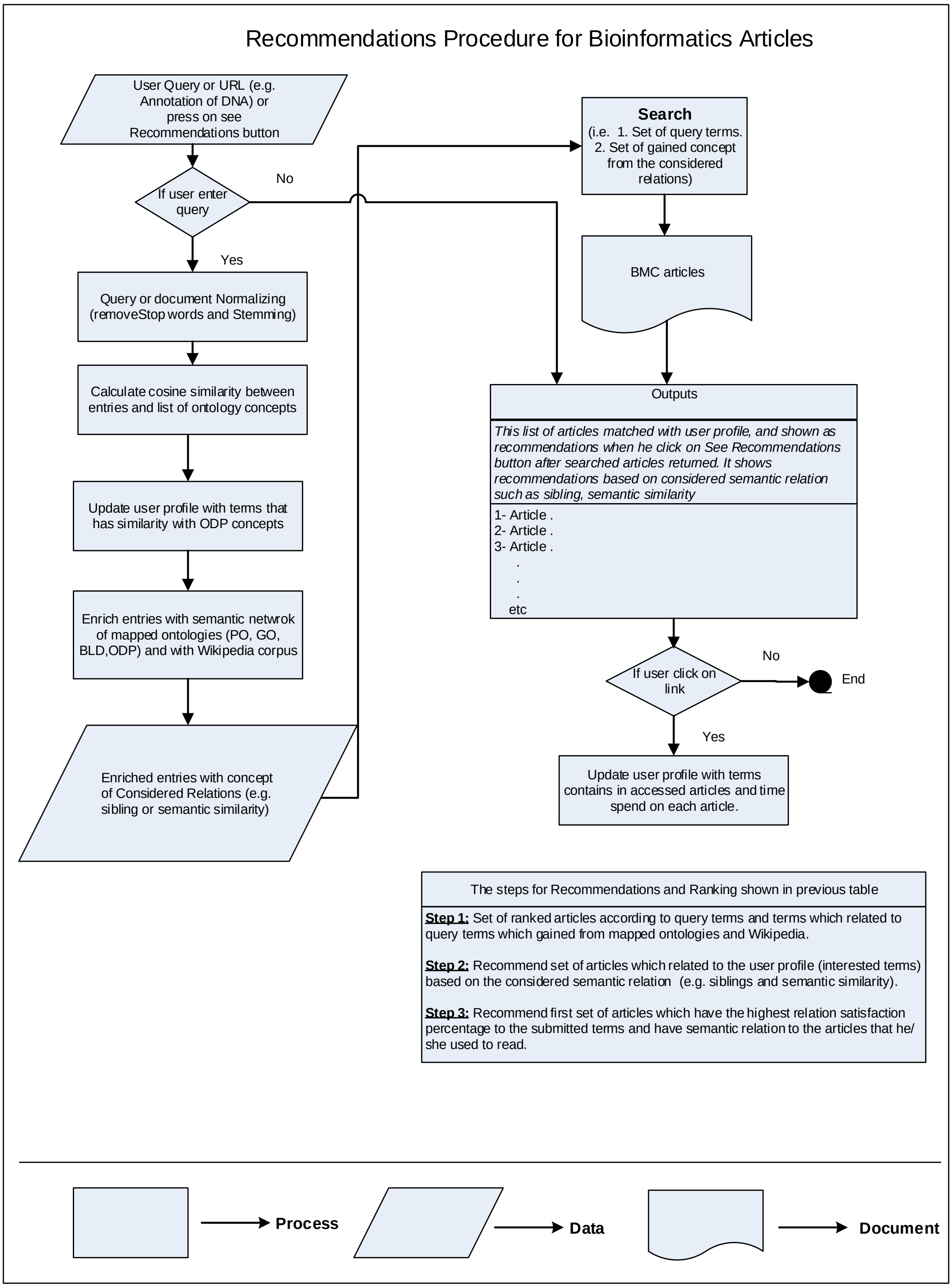

Wikipedia Corpus in OWL
Bioinformatics ontologies (e.g. GO and PO)
Bioinformatics Websites ontologies (e.g. ODP and BLD)
Domain Specific Resources
Extract semantics Relations and associations
Search for user's query after enriched with exploited relation concept
Return articles
BMC Corpus
Prototype System
Submit a query/Browse recommendations
Shows results on searched articles and updated recommendations on the preferred articles
User

Figure 3.3: Specialist Search Framework.

Furthermore, figure 3.4 shows the sequence of tasks that our approach will take to provide users with most relevant recommendations. It explains the steps that were taken in order to post a query to our system and get recommended articles that will be organised based on the user profile. Also, it shows the steps that will be undertaken in order to classify the browsed contents to the ODP ontology concepts. So, if the user submitted a query, then his/her query will be normalised (i.e. remove stop words, stem, etc.). After that, it will be matched with the most similar concept from the ODP concept. Then, it will be enriched with concept that has semantic relation (whether sibling or semantic similarity) with the ODP concept. Then, it will be searched in the BMC corpus. After that, a link will be shown with the retrieved results that allows the user to check his/her recommendations. Otherwise, if the user checks the recommendations without submitting a query, then the results will be shown based on his/her preferences as well as the exploited semantic relation (sibling or semantic similarity).


Recommendations Procedure for Bioinformatics Articles
User Query or URL (e.g. Annotation of DNA) or press on see Recommendations button
If user enter query
No
Yes
Query or document Normalizing (removeStop words and Stemming)
Calculate cosine similarity between entries and list of ontology concepts
Update user profile with terms that has similarity with ODP concepts
Enrich entries with semantic netwrok of mapped ontologies (PO, GO, BLD,ODP) and with Wikipedia corpus
Enriched entries with concept of Considered Relations (e.g. sibling or semantic similarity)
Search
(i.e. 1. Set of query terms. 2. Set of gained concept from the considered relations)
BMC articles
Outputs
1- Article .
2- Article .
3- Article .
.
.
.
etc
If user click on link
No
End
Yes
Update user profile with terms contains in accessed articles and time spend on each article.
Process
Data
Document

This list of articles matched with user profile, and shown as recommendations when he click on See Recommendations button after searched articles returned. It shows recommendations based on considered semantic relation such as sibling, semantic similarity

The steps for Recommendations and Ranking shown in previous table

Step 1: Set of ranked articles according to query terms and terms which related to query terms which gained from mapped ontologies and Wikipedia.

Step 2: Recommend set of articles which related to the user profile (interested terms) based on the considered semantic relation (e.g. siblings and semantic similarity).

Step 3: Recommend first set of articles which have the highest relation satisfaction percentage to the submitted terms and have semantic relation to the articles that he/ she used to read.

Figure 3.4: Recommendation Procedure for Bioinformatics Articles.

3.2.3 System Users and Profiles

The user who is shown in figure 3.1 is the bioinformatics specialist, who represents the main target whose needs our prototype recommender system is intended to cover. Each user has his/her own profile, which helps our method to determine user preferences based on browsing history. As discussed in the previous chapter, ontologies can be used to represent user profiles. Thus, our prototype system uses ODP ontology to represent the user profile due to the ontology's efficiency and accuracy in providing a user profile that riches with semantics. These semantics can be exploited to enhance the accuracy of the provided recommendations.

Thus, constructing ontological user profiles requires several data pre-processes in order to offer each user individual recommendations based on his/her preferences or interests. Therefore, determining the needed information is an important task to start constructing our user profile. Our approach will collect user data from three different entries, namely, surfed URLs, clicks and bookmarked webpages. For instance, the browsed URLs will help us to determine the preferred topics of the user and they will be used when constructing the ontological user profile to match the URL with the most similar concept of ODP ontology, which will be used to represent our ontological user profile. Then, the clicks show which browsed URL is important to the user or not, since the increase in the number of clicks means that a URL is important and vice versa. After that, bookmarks support the importance of the browsed URL to the user, so whenever a user bookmarks a URL, this means that a bookmarked URL is important to him/her.

Moreover, an automatic method is in charge of adding, updating and deleting interests from the profile and these are aimed to keep user profile up-to-date and allow him/her to receive updated recommendations tailored to his/her user profile. Figure 3.5 illustrates the data-flow diagram for formulating an ontological user profile that supports our recommender approach in order to provide more accurate recommendations tailored to each individual user based in his/her preferences. The following sections describe how terms are extracted to support our resources and the method that can be used to calculate term frequency in the text and the steps that were taken to classify surfed URLs into the ODP ontology.

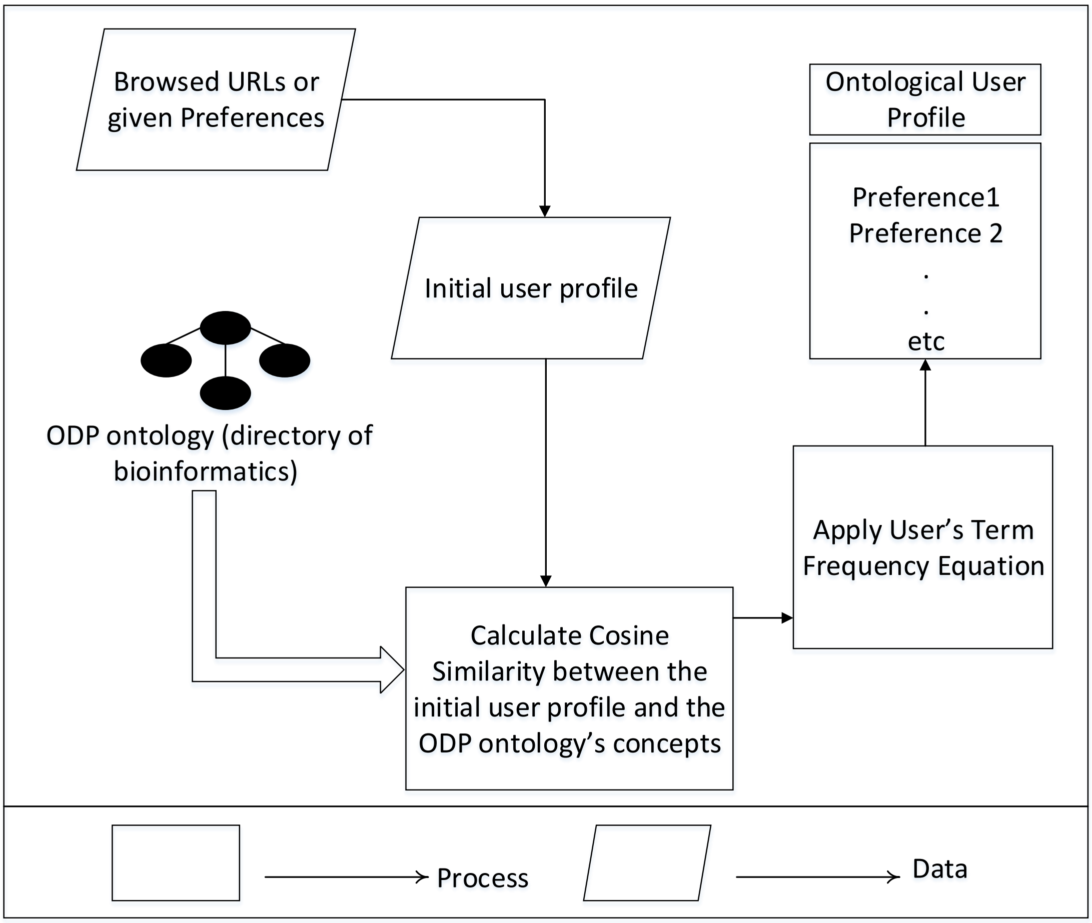

Browsed URLs or given Preferences
Initial user profile
Ontological User Profile
Preference1
Preference 2
.
.
etc
ODP ontology (directory of bioinformatics)
Apply User's Term Frequency Equation
Calculate Cosine Similarity between the initial user profile and the ODP ontology's concepts
Process
Data

Figure 3.5: Data Flow Diagram for formulating an ontological User Profile.

3.2.3.1 Bioinformatics Terms

These terms are important because they reflect the domain that we are concerned with and they are used to compare entries (i.e. surfed URLs, clicks and bookmarked webpages). So, preparing these terms by calculating their Term Frequency (TF) and Term Frequency-Inverse Document Frequency (TFIDF) [125] will reduce the time and effort needed to perform several calculations such as a new term classification. These terms were extracted from the ODP (i.e. terms under bioinformatics concept), BLD, GO and PO ontologies, totalling approximately 400 terms that randomly selected and retrieved from the Wikipedia corpus (i.e. formulated as a set of concepts in OWL file for reasoning purposes).

3.2.3.2 Term's Weight

There are multiple ways to calculate a term's weight, and these methods differ in their accuracy. TFIDF [125] is the most popular method for calculating the term weight, and it has been used in many approaches. So, in order to calculate the term weight in a document, two steps need to be completed. Firstly, term frequency in the document must be calculated according to the equation below:

$$TF(t_1, d_1) = \frac{Number\,of\,t_1\,in\,d_1}{Total\,Number\,of\,terms\,in\,d_1} \tag{3.1}$$

Where TF represents term frequency in a document, t_1 represents the term in the document and d_1 is the document that contains t_1 .

We then use the following equation to calculate the inverse-document frequency:

$$IDF(T) = log\frac{\mid D \mid}{\mid d : t \in D \mid} \tag{3.2}$$

Where IDF represents the inverse-document frequency, D represents the entire collection and d is the number of documents that contain term t.

Thus, all previous equations are used to calculate TFIDF, which reflects the term's weight as follows:

$$TF(t_1, d_1) * IDF(t_1) \tag{3.3}$$

3.2.3.3 Term's Classification

Several methods can be used to classify users' entries (i.e. surfed URLs or given preferences) with ontologies, and by using these methods, any URL can be classified into the most appropriate concept in the used ontology. For instance, when a user frequently browses websites that discuss concepts such as DNA, RNA, proteins and so on, then the classification method will map these concepts with those stored in the ontology. Then, the importance of each concept will be calculated for each user by considering different factors such as number of visits and time spent on the page. Cosine similarity [124] is a classification method applied in this approach in order to classify each surfed URL to the most appropriate concepts. Two vectors are required in the classification process. The first vector is represented by user entries, which are URLs, and the second vector is the concept's description that is stored in the ODP ontology. The following equation is used to calculate cosine similarity between these two vectors:

$$CosineSim(d_1, d_2) = \frac{\sum_{i=1}^{n} d_1 * d_2}{\sqrt{\sum_{i=1}^{n} d_1 i^2} * \sqrt{\sum_{i=1}^{n} d_2 i^2}} \tag{3.4}$$

Where d_1 represents the first vector (i.e. surfed URL) and d_2 represents the second vector, which is the concept's description. A set of pre-processes have to be completed to calculate the cosine similarity between both vectors. Firstly, we remove stop words and return each concept to its stem and then calculate the cosine similarity for each concept in the first document with all concepts in the second document. Finally, each concept is mapped to the concept that satisfies the highest cosine similarity, where a cosine similarity of 0 means no similarity between the compared concepts and a score of 1 indicates that the compared concepts are the same.

3.2.3.4 User profile for Sibling Experiment

This section will discuss formulating the user profile for the experiment 6.3, which assesses the effectiveness of applying our developed sibling method on the provided recommendations.

The adaptive ontological user profile method supports our recommender method, since it allows the recommendation method to provide tailored recommendations to each bioinformatician based on his/her preferences. Moreover, it allows the recommended items to be shown to the specialist based on their assigned weights (i.e. *Term Frequency*), which represent each item's priority to the bioinformatician. The interest with the highest weight will be shown at the top of the results that appear for him/her. Furthermore, to make the created user profile adaptable, there is a mechanism that allows our user profile to be updated for each user, since it has been equipped with add, delete and update methods that can be run automatically.

For instance, suppose a bioinformatician is interested in DNA, RNA and protein and then visited a website that discussed DNA, where he/she spent 5 minutes reading an article. After that, he/she visited another website that discussed the second preferred topic, which is RNA, and spent 3 minutes there. Then, he/she visited a website that discussed his/her third preferred topic, where he/she spent 1 minute. The following steps will briefly describe the process of creating an adaptive ontological user profile for the bioinformatician.

1. All information, such as URLs visited, time spent at each site, clicks and bookmarks will be collected by a browser plug-in installed at his/her machine.

2. Then, to construct an ontological user profile, the cosine similarity between the surfed URL and ODP concept will be calculated. Its score will be aggregated with the details collected in Step 1 to formulate the *Term Frequency* equation 5.1.

3. After that, for the user profile adaptation, a method will be used to update and delete users' preferences daily, based on a threshold that is tuned for daily increases and decreases in the preferences' priority (i.e. *Term Frequency*) for the user.

This represents an abstract view for all of the main steps that our method will follow to construct an ontological adaptive user profile. This mechanism and all implementation steps will be discussed in detail in the implementations in chapter 5.

To this end, the bioinformatician will have a user profile that contains the three preferred topics, and each of which will be assigned with weights that reflected their priority to him/her. Then, the bioinformatician's profile will be mapped with the inferred semantic network to exploit the relations and information gained from multiple resources (this will be discussed in detail in Section 4.6, and its implementation will be discussed in more detail in Section 5.3.3). Finally, through this profile, the bioinformatician will have articles recommended to him/her that have a semantic relation to his/her preferences based on the exploited relation (i.e. sibling or semantic similarity) from the inferred semantic network.

3.2.3.5 User profile for Semantic Similarity experiment

This section will discuss formulating the user profile for the experiment which assesses the effectiveness of applying our developed semantic similarity method. This method contributes to more the enhancement of the accuracy of the provided recommendations than the method which exploits sibling relation to enhance the accuracy of the recommendations. However, this experiment has not followed the same method of formulating an ontological user profile as the experiment which exploited sibling relation to provide recommendations. Thus, the difference between this method and 3.2.3.4 method is the way of collecting user preferences. In this method, user preferences were collected explicitly, not like the previous method, which collected preferences implicitly.

3.3 Recommendation Service

This research contributes to developing semantic-based methods for identifying relations and hidden associations extracted from bioinformatics resources (e.g. ontologies such as PO, GO, ODP and BLD and corpora such as Wikipedia). We have studied the aforementioned resources and have concluded that implicit information can be extracted through semantic analysis. Our central hypothesis is that this can be used in providing better recommendations to specialists in specific domains such as bioinformatics. In addition, we automatically tailored the recommendations to the specialists' needs based on their profiles by collecting specialist preferences and interests implicitly. We aim to demonstrate our methods by providing recommendations to bioinformaticians on the most relevant contents (i.e. articles) from the BMC.

This recommendation service method is used as a prototype system that exploits the semantic relations gained as a result of the inference process that has been undertaken over the overlapped information among multiple bioinformatics resources. Specifically, it exploits sibling and semantic similarity relations by enriching the specialist's query to retrieve data from the BMC croups. Then, it recommends articles that have sibling or semantic similarity relations with the articles that the specialist researcher has read previously or to his/her preferences stored in the user profile.

Our task to make this recommendation service method available consists of two branches: (i) exploiting the developed methods and extracted semantic information, such as sibling relations, from multiple resources (e.g. ontologies, taxonomies, Wikipedia) and reasoning with this information to obtain more useful information that can help to provide better recommendations; and (ii) constructing an ontological user profile based on information extracted implicitly from the user-surfed sessions and interaction with the system (this will be covered in more detail in the chapter 5). The information from (i) and (ii) is then combined to enrich the specialist query and provide more accurate recommendations to specialists in the domain of bioinformatics.

In order to achieve these targets, we first developed a reasoning method to exploit overlapping information between different bioinformatics resources, such PO, GO and extract semantic relations and hidden associations between different classes. This method used SPARQL[1] queries to extract information and provided them to the reasoner, which combined them with semantic rules to infer new relations that may exist among resources (this will be discussed in detail in section 4.2 and its implementation will be discussed in section 5.4). As a result, a semantic network was created which represented the extracted semantic relations and hidden associations from the intersection between different resources. This includes new identified relations not found in the original resources (this will be covered in more detail in the chapter 5). Users' profiles were then boosted by adding the relevant information from this network.

The profile that was used in our approach is able to accommodate the frequent changes of the user preferences and the enrichment with valuable information gained from the semantic network and provided fully automated solutions (this will be discussed in more detail in the next chapter). This approach was fully automated and tailored recommendations to each user individually based on his/her preferred topics.

[1] http://www.w3.org/TR/rdf-sparql-query/query

3.4 Evaluation Methods

The evaluation methods in recommender systems differ from one approach to another based on their intended purpose. Shani and Gunawardana [189] classified three methods that can be applied to recommender systems for evaluation purposes. These are offline evaluation, user studies and online evaluation. The offline evaluation method is undertaken using pre-collected data that represent a sample of the real environment and does not require any interaction from the system's user. This provides an opportunity to authenticate its results and consider them as initial and not final results because it sometimes shows a huge difference when evaluating real data. However, this type of evaluation may not be as accurate. But, the advantage of applying such a method is the low cost. This evaluation method was applied to different recommender systems such as [190] and [191].

The user studies method, which differs from the previous method, is based on user interaction. Such an approach uses a set of tasks that should be performed by a real user [189]. The mechanism behind this method is observing users' behaviours and collecting their interactions while they perform the provided tasks. Such a method is widely used through approaches; for example, as seen in [192], [193], [194] and [195]. Yet this method presents some drawbacks that may complicate evaluations. Such user studies are expensive to undertake in terms of effort and time. There are difficulties with collecting a large number of volunteers or candidates to increase the precision of the returned results for this type of evaluation. Also, there is a difficulty in controlling a large group of participants to make them follow specific procedures, and such a problem can weaken this method [189].

Lastly, online evaluations are undertaken in real time in a real environment with real users. The users' behaviour is observed, and their interactions are collected while they use the system. Different approaches can be compared, and one can monitor the change in their behaviours when they interact with different recommender systems. This method is highly precise and reliable since it reflects a real interaction between users and systems. In addition, such a method was employed by different recommender systems that attempt to provide recommendations that were evaluated accurately and precisely, especially huge systems that have a large number of users. Researchers have also considered using this type of evaluation mechanism in their systems such as [196], [197] and [198]. However, this method has some restricting limitations. These drawbacks include the high cost of publishing a system that will consume time and effort during evaluation. Also, the disparity in the precision between system parts has negative effects on provided recommendations because such a problem will not be enhanced before it appears to the user. Furthermore, Gunawardana and Shani [199] suggest that incorrect or inaccurate recommendations may cause some users to stop using the system and use others since they may be unhappy or have discovered weak points in the system after conducting an online evaluation.

We intend to undertake a user study evaluation due to its flexibility in allowing participants to interact with the system; it will be used to assess our developed content-based recommendation method in the field of bioinformatics. It will give us an opportunity to examine the accuracy and relevance of the produced recommendations. An online and offline evaluation, in contrast, the former would be too expensive to use as part of our evaluation process and for the purpose of this PhD project. The latter would be not accurate enough to asses the accuracy of the provided recommendations. Moreover, offline evaluation needs for pre-pared dataset in order to perform the evaluation, but we do not have a pre-pared dataset for the bioinformatics.

3.5 Evaluation Metrics

Different aspects of our approach will be combined together as single approach, then compared with the other systems. Herlocker et al. [200] explained that evaluation metrics are divided into the following three types:

- *Predictive accuracy metric.*
- *Classification accuracy metric.*
- *Rank accuracy metrics.*

Each of these will be described in more details below.

3.5.1 Predictive Accuracy Metric

It is a useful type of non-binary rating where the result will be shown based on its priority to the user profile. Since it rates the items with the highest rates related to the user. It can be used in several fields that concentrate on recommending items such as documents, movies and music. So, the essential representatives of this metric are mean absolute error (MAE), mean squared error (MSE), root mean squared error (RMSE) and normalised mean absolute error (NMAE). The MAE is defined as follows:

$$MAE = \frac{\sum_{i=1}^{N} \mid p_i - r_i \mid}{N} \tag{3.5}$$

where N is the number of items, p_i is predicted items or true items and r_i represents all rated items. The MAE calculates the average error between all rated items and items rated by the user.

The MSE is calculated as follows:

$$MSE = \frac{\sum_{i=1}^{N} (p_i - r_i)^2}{N} \tag{3.6}$$

The RMSE is calculated as follows:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N} (p_i - r_i)^2}{N}} \tag{3.7}$$

Both MSE and RMSE metrics were designed to calculate the error average such as MAE; however, these two metrics were designed to emphasise calculating larger errors [201]. Moreover, the normalised mean absolute error (NMAE), which is the fourth representative of predictive accuracy metric, normalises the MAE in order to make the result comparable among recommender systems with various rating scales [201].

3.5.2 Classification Accuracy Metric

It is a useful type for measuring the correct and incorrect level and whether the recommended items are relevant. It is not concerned with the ranking of recommended items that occur based on their priority. Classification accuracy metrics include precision, recall, fallout, miss rate, inverse precision and inverse recall

[201]. Each was designed to calculate a specific thing in order to classify an evaluated item. The following table, illustrated in [201], shows the possible cases that classification accuracy metrics can have:

	Relevant	Irrelevant
Recommended	tp (true positive) Correct result	fp (false positive) Unexpected result
Not Recommended	fn (false negative) Missing result	tn (true negative) Correct absence of result

Table 3.1: Classification Accuracy Metric

Precision involves calculating the ratio of recommended items that are relevant to all recommended items. Equation 3.8 reflects the probability that recommended items are related to the user interests:

$$Precision = \frac{tp}{tp + fp} \tag{3.8}$$

Recall involves calculating the percentage of the correct recommend items over the total available relevant items. This measure represents the probability that relevant items were recommended. Moreover, precision and recall have an inverse relation, so when the value of precision increases, the value of recall will decrease and vice versa. Equation 3.9 shows how to calculate the recall metric:

$$Recall = \frac{tp}{tp + fn} \tag{3.9}$$

Fallout involves calculating the fraction of the recommended items irrelevant to the whole irrelevant items. This measure can provide the probability that irrelevant items are recommended, and so equation 3.10 shows how to calculate the fallout indicator:

$$Fallout = \frac{fp}{fp + tn} \tag{3.10}$$

missRate involves providing the percentage of items relevant to the total number of relevant items that however, have not been recommended. Equation 3.11

shows the probability of the existing relevant items, but they have not been recommended:

$$missRate = \frac{fn}{tp + fn} \tag{3.11}$$

inversePrecision involves calculating the ratio of items irrelevant to the non-recommended items, and they have not been recommended by the recommender approach. So, equation 3.12 can provide the probability of irrelevancy of items that are not recommended:

$$inversePrecision = \frac{tn}{fn + tn} \tag{3.12}$$

inverseRecall involves calculating the percentage of items not recommended, and they are irrelevant to the total irrelevant items. So, equation 3.13 shows how the probability of irrelevant items that are not recommended is calculated:

$$inverseRecall = \frac{tn}{fp + tn} = 1 - Fallout \tag{3.13}$$

Furthermore, Shani and Gunawardana [189] suggested that precision to recall or true positive rate to false positive rate curves can be used on the number of recommended items when the number presented to the user is not determined. **Precision** at **N** and **Mean Average Precision** [202] are most useful for recommender systems [203], especially when the recommended items presented to the user are preordained [204].

3.5.3 Rank Accuracy Metric

It is differ from the previous metrics, and concerned with ranking the retrieved items. Such metric is not concerned with the accuracy or rating prediction, so the highest relevant items will be placed at the top of the retrieved results, and those items that are less relevant will be placed at the bottom of the retrieved results. This metric is useful for search engines because it ranks the retrieve results based on their priority to the user [201]. Arzanian et al. [205] illustrated a method for measuring the rank accuracy, which compares the average ranking performed by some proposed algorithms and the user ranking, and so equation 3.14 can be used to calculate such metrics:

$$RankAccuracy = \frac{1}{m} \sum_{i=1}^{m} \mid r_i - r'_i \mid \tag{3.14}$$

where m represents the retrieved items, r_i is the user ranking items and r'_i is the items ranked by the proposed systems or algorithms.

Each of the aforementioned metrics was designed for specific purposes. The evaluation metric can be selected based on several issues, such as the type of approach being evaluated and the method of representing the items or results. Our recommender approach (in the bioinformatics field) will provide a list of articles, so due to the nature of the provided contents, it needs to consider two types of metrics. The first type is a classification accuracy metric, which is very useful for recommender systems. **MAP** and **Precision** at **N** will be used to assess the classification accuracy of our approach. The second type is a predictive accuracy metric, which is also useful for recommender systems to assess the level of predictive accuracy of our recommender approach compared with other comparative approaches. Thus, the **MAE** metric will be considered for accuracy assessment of the predictive metric.

Our recommender approach will be compared with other related approaches, where the comparison between our and their approaches will be conducted with regards to the parts that share the same functionalities. However, the user (who is selected to assess our content-based method, which uses user-centric evaluation) will have the same frontend, and she/he will not be aware that she/he is dealing with different approaches that provide the same service. After this comparison between the different approaches, with some shared functionalities, has been conducted, we will have a better idea of the accuracy that can be achieved with our recommendation approach. If our approach achieves the highest result, this will indicate that our assumption has been successfully demonstrated. Otherwise, we will need to identify the factors in our recommender approach that have led to the inadequate performance and subsequently conduct further work or adjust it.

3.6 Conclusions

This chapter discussed the architectural framework that our approach used to support a specialist domain with accurate recommendations by exploiting semantic relations and overlapping information between multiple resources and search activities. Then, providing these recommendations to each bioinformatics specialist based on his/her preferences. Moreover, it discussed various resources that we have considered in this work and the format that all of our processed resources should be in. This common format is needed to allow our reasoning method to go through them and extract semantic relations and hidden associations that would not exist when we looked for them in a single resource. Furthermore, it discussed the methods that applied to extract information form the Wikipedia and the steps for converting data from text format to be as a set of concepts in OWL format. Then, the chapter discussed the user profile conceptually and considered the different requirements that should be aggregated in order to create an adaptive ontological user profile. This is to help our approach support each bioinformatician with recommendations based on his/her preferences. Moreover, it discussed the recommendations method as an implementation method for the contributions of this research. This method provides recommendations to the specialist in the domain of bioinformatics, which is our intended target that need to be enhanced by exploiting semantic relation and hidden associations between different bioinformatics resources. Finally, it discussed the evaluation methods that will be used to assess the recommender system and its performance to measure the level of enhancement that could be achieved when applying our discovered semantic relations into our specialist recommender system in the bioinformatics field.

Chapter 4

Semantic-based Techniques for Specialist Search

4.1 Introduction

This chapter introduces the main contributions of this research and discusses them from a conceptual perspective. The aim of this work is to develop a semantic method that supports specialist users in their searches and provides them with accurate recommendations based on their preferences and search activities. This is done by developing several methods to satisfy the main goal of this research.

1. We develop a semantic method that is able to reason through bioinformatics resources and then extract different semantic relations and hidden associations between multiple bioinformatics resources which may have different structures. The method is able to handle them even though they have different structures, nature and relations. Then, it infers a new type of relation that cannot be found when the specialist just searches in a single resource; such a novel relation is the sibling relation. After that, it employs them to enhance the accuracy of the recommendations provided.

2. We introduce seven reasoning rules to fire during the reasoning process and discover several semantic relations that may exist between the processed data as a result of the information overlapping between multiple resources. These

rules contribute to discovering new information and associations which can be exploited by our semantic method to infer further new data that support our recommender approach with valuable information to enhance the accuracy of the recommendations provided.

3. We devise a method that represents the inferred relations and associations in the form of an inferred semantic network. This is used to enrich the user profile with information drawn from multiple resources and represented in the inferred semantic network. The inferred semantic network is an essential stage that is needed to give our semantic method the opportunity to exploit all inferred relations and information. It supports our recommender approach with valuable information that can be used to provide more accurate results. This method can be distinguished by its ability to overcome most challenges that could be faced when dealing with multiple resources with varying structures and different relations. It supports the other methods in our approach, such as the user profile enrichment with the required relations (whether a sibling or semantic similarity relation). Moreover, it is supported with a method to update parts of the inferred semantic relations, which helps to keep the inferred data up to date for supporting the specialist with accurate recommendations.

4. We develop a semantic similarity method which reasons through different bioinformatics resources and then calculates the concepts' description similarity and semantic similarity between concepts during the inference process. It employs a semantic custom rule (the seventh rule of our reasoning rules) to decide whether two concepts within the same resource or from different resources are semantically similar or not. It contributes to finding new information that would not be found by using the six semantic rules that run over the reasoning process, since it works as a complementary method to the seventh semantic rule (semantic similarity rule) to find semantic similarity cases between different concepts during reasoning process. This discovered information can help the specialist researcher to have better results and more accurate recommendations.

Also, it discusses the method that we developed to enrich the profile with the inferred semantic relation gained from the overlapped information among different bioinformatics resources.

4.2 Semantic Methods

The availability of information on the World Wide Web (WWW) is continuing to grow rapidly, including specialist resources such as corpora and repositories of information, so the task of extracting valuable information can be more challenging than before, especially for unconnected resources. Although semantic-based methods can help alleviate this problem, this still presents a challenge as resources may have varying structures. Semantic-based techniques are required to infer new information that may not be found in the original resources, specifically in fields that have different resources and various structures, such as bioinformatics. This semantic information can be exploited for several purposes. For instance, in the field of bioinformatics, this information can be used to discover drugs and search for and extract related information more accurately and efficiently. Our work aims to develop semantic-based techniques that exploit relations and associations within and across different resources to provide recommendations on the content of interest (i.e. articles). It helps the bioinformatician by providing him/her with the most relevant content based on his/her preferences and inferred data.

Although, some relevant works have tried to exploit the overlapping and complementary information to provide recommendations based on the inferred data, these works still suffer from some limitations in discovering and employing the determined relations and information. For instance, Mirizzi et al. [159] provided a recommender system for movies based on the LOD dataset. They used semantic information gained from this ontology; however, the inferred triples are not employed as effectively to discover new relations and associations from the overlapping information. Furthermore, they did not consider using ontologies when formulating the user profile; also, considering an ontological user profile may help to enrich such profiles and help enhance the accuracy of the provided recommendations. Thus, these drawbacks may result in the provided recommendations lacking in accuracy.

Therefore, our semantic-based methods will attempt to overcome these shortcomings that appeared in the other approaches. Moreover, our work contributes in supporting researchers in specific domains such as bioinformatics with accurate recommendations gained from the semantic relations (i.e. sibling and semantic similarity relations) which are inferred from the overlapped information among different resources. This method is distinguished by its ability to handle multiple resources with various structures. It is also supported with semantic rules that are able to extract specific types of relations and then employ them in the recommendation process to enhance the precision of the recommendations provided.

4.2.1 Semantic Rules Definition and Analysis

This thesis provides seven semantic rules that have been discovered after studying and observing the different relations included in our different processed resources. Each rule has a target which tries to discover when it runs through the processed resources. The target of each rule can be satisfied by its ability to discover the needed data, measure its efficiency when this rule applies over our dataset. Then, determines how much enhanced information the bioinformatician gained when the semantic relation is applied in our prototype recommender system. Therefore, in this section, we will define each rule and clarify each rule's main target. In Section 4.2.3, we will provide further analysis of each rule and an example which can illustrate its efficiency.

Table 4.1: Semantic Rule Terminologies.

Semantic Rule Terminologies
Classes := C_1,..., C_n
C :=name, subClassOf, Comment, label,equivalentClassOf,objectProperty
subClassOf := C ∈ Classes
Restrictions := onProperty,objectProperty,SomeValuesFrom
onProperty := C ∈ Classes
SomeValuesFrom := C ∈ Classes

Rule 1 SuperClassOf: This rule allows parents to recognise their children,

since our resources only have the opposite relation, where only the child recognises its parents. Considering this fact will help enhance the recommendations, since this relation allows the bioinformatics researcher to have all possible details about his/her preferred concept. Moreover, this relation helps construct our inferred semantic relation and hidden association in a semantic network easily and effectively. This will allow our prototype system to exploit such relations to enhance the accuracy of the provided recommendations. The formal definition of this relation is presented in Table 4.2

Table 4.2: SuperClassOf Rule.

SuperClassOf Rule

Requirement: *List of InfModel*

Input: *Selected Data (e.g. class, subclassOf, etc.)*

Output: *List of inference triples*

Rule (SuperClassOf): this rule shows that x is parent & superclass of y :

SuperClassOf(x,y) $\Longrightarrow$ x $\in$ Classes $\wedge$ y $\in$ Classes $\wedge$ x $\in$ y.subCLassOf

Rule 2 GrandSubClassOf (transitive): This rule allows concepts to be aware of their grandparent or grand ancestor and can help enhance the recommendations. It gives the bioinformatician the opportunity to obtain recommended articles that have any relation with his/her preferred concept when this inferred relation is exploited by our prototype recommender system. Table 4.3 illustrates the formal definition of this rule.

Table 4.3: GrandSubClassOf Rule.

GrandSubClassOf Rule

Requirement: *List of InfModel*

Input: *Selected Data (e.g. class, subclassOf, etc.)*

Output: *List of inference triples*

Rule (grandSubClassOf): this rule shows that x is grand Child of y :

grandSubClassOf(x,y) $\implies$ $x \in \text{Classes} \land y$

$\in \text{Classes} \land \exists\, z\ (z \in \text{Classes}) \land$

$z \in \text{x.subClassOf} \land y \in \text{z.subClassOf}$

Rule 3 Sibling: This rule allows researchers to discover all concepts that have the same parent, even if they exist among different resources. Since each resource has its own way to describe concepts and since some information (e.g. children) may be mentioned in some resources but not others. Thus, applying this rule will allow us to be aware of all children (i.e. concepts) that are related to the same parent, even if they are not in the same resource. Moreover, exploiting such relations will support our prototype system and help researchers by providing them with the most accurate recommendations regarding articles that discuss concepts which are relevant to their preferred concepts. This may broaden their horizons with extra knowledge about their interested topic. Table 4.4 provides the formal definition of this rule.

Table 4.4: Sibling Rule.

Sibling Rule

Requirement: *List of InfModel*

Input: *Selected Data (e.g. class, subclassOf,etc.)*

Output: *List of inference triples*

Rule (Sibling): this rule shows that x is sibling of y :

sibling(x,y) $\Longrightarrow$ x $\in$ Classes $\wedge$ y $\in$ Classes $\wedge$

$\exists$ z (z $\in$ Classes) $\wedge$ z $\in$ x.subClassOf $\wedge$ z $\in$ y.subClassOf

Rule 4 Is_type_Of: This rule helps parents in our dataset gain their children's types (i.e. class, property, etc.). This will support our prototype recommender system by illustrating each concept for the bioinformatician from different perspectives, which may help him/her discover all possible facts about his/her preferred concept when these facts are inferred from the preferred concept's children. Table 4.5 shows the formal definition of this rule.

Table 4.5: Is_type_Of Rule.

Is_type_Of Rule

Requirement: *List of InfModel*

Input: *Selected Data (e.g. class, subclassOf, etc.)*

Output: *List of inference triples*

Rule (Is_type_of): this rule shows that x can be of type y :

Is_type_Of(x,y) $\Longrightarrow$ x $\in$ Classes $\wedge$ $\exists$

z (z $\in$ Classes) $\wedge$ x $\in$ z.subClassOf $\wedge$ y $\in$ z.type

Rule 5 SameAs: This rule allows researchers to discover whether a concept is mentioned in any other resource under a different name. This is accomplished by comparing concepts' descriptions and included details. This will help our prototype recommender system show different names that could be used to name a concept, which will help the researcher gain extra knowledge about the concept. Moreover, it may help researchers to discover other relations and information mentioned in one resource but not others. Table 4.6 provides the formal definition of this rule.

Table 4.6: SameAs Rule.

SameAs Rule

Requirement: *List of InfModel*

Input: *Selected Data (e.g. class, subclassOf, etc.)*

Output: *List of inference triples*

Rule (sameAs): this rule shows that x sameAs y :

sameAs(x,y) $\implies$ x $\in$ Classes $\wedge$ y $\in$ Classes

$\wedge$ x $\neq$ y $\wedge$ x.description $\simeq$ y.description

Rule 6 Equivalent: This rule allows researchers to discover new facts by equalizing two concepts when the concept preferred by the researcher has *onProperty* and shares *someValues* with the other concept. Then, these two concepts are equalised, even if they are located in different resources. This will support our prototype recommender system in exploiting the equivalence between these concepts and providing extra information and recommendations to the researcher, which are gained as a result of this equivalency. Because, each concept has its own descriptions, relations and associations which can be exploited and provided to the researcher to discover new articles that are relevant to the article that discusses the concept that he/she used to read about it. Table 4.7 provides the formal definition of this rule.

Table 4.7: Equivalent Rule.

Equivalent Rule

Requirement: *List of InfModel*

Input: *Selected Data (e.g. class, subclassOf, etc.)*

Output: *List of inference triples*

Rule (equivalent): this rule shows that x equals y :

$\text{equivalent(x,y)} \implies \text{x} \in \text{Classes}$

$\wedge\ \text{y} \in \text{Classes} \wedge \text{z} \in \text{TransitiveProperty} \wedge \text{x.type} = \text{Restrictions}$

$\wedge\ \text{z} \in \text{x.onProperty} \wedge \text{y} \in \text{x.SomeValuesFrom} \wedge \text{y} \in \text{z.Part\_Of}$

Rule 7 Semantic Similarity: This rule allows researchers to discover any semantic similarity between a specific concept and any other concept in the same resource or other resources. The semantic similarity relation will be considered or decided when the similarity score between the concepts exceeds all suggested thresholds. Thus, we can say that two concepts have a semantic similarity relation when they have similarities based on two factors: their description and position in their resources. This semantic similarity rule fires and is calculated during the reasoning process. This gives an opportunity to semantic similarity method to find more semantic similarity cases that may not appear when calculating the semantic similarity as it is calculated traditionally (which is based on calculating the semantic similarity without considering the concept description and during the inference process). This rule will support our recommender system in providing more accurate articles that are relevant to the concept preferred by the researcher. Table 4.8 illustrates the formal definition of this rule.

Table 4.8: Custom Built-in Semantic Similarity Rule.

Custom Built-in Semantic similarity Rules

Requirement: *List of InfModel*

Input: *Selected Data (Classes and Classes' Comments)*

Output: *List of concepts*

Rule (Semantic Similarity): This rule shows that x has a semantic similarity relation with y

similar(x,y) $\Longleftarrow$ x $\in$ Classes $\wedge$ y $\in$ Classes $\wedge$ x $\neq$ y $\wedge$

$\exists$ a (a $\in$ x) $\wedge$ $\exists$ b

(b $\in$ y) $\wedge$ similar(x,y,a,b) = true

In order to infer or discover semantic relations between different concepts in our processed dataset, we have taken each concept and passed it through all other concepts in our dataset. We then checked the shared properties or relations among different concepts that could lead us to conclude or infer any semantic relation that may help us enhance the accuracy of the provided recommendations. This can be summarised in the following steps:

1. Apply a SPARQL query and retrieve all contained classes in the processed data.
2. Take each class and its content and compare it to the other classes to confirm all existing relations between the compared pairs.
3. Observe the compared concepts and draw any new relations that can be inferred between them (this point will be expanded under each inferred relation, and we will discuss how we discovered each relation and what the developers should do in order to reach the discovered relations in their dataset).
4. Name the inferred relation based on the type of relation inferred. For instance, sibling relations have been called "sibling", since this type of relation

tried to discover siblings from multiple resources and exploit them to provide recommendations.

5. Test the inferred relation and apply it to different datasets, such general data sets related to humans, cars, foods and so on. Then, test the inferred relation on the specialist dataset (in our case, we applied our inferred relation to bioinformatics data).

6. Evaluate the inferred relation and measure its effectiveness over our dataset and how much it enhanced our prototype recommender system.

For the last step (6) in this thesis, we only evaluated the most promising relations, which are sibling and semantic similarity. This was done by integrating these two relations into our prototype recommender system and comparing the performance of the recommender system with and without considering these relations. Also, these two relations were compared against each other, other relevant work from the literature and general recommender systems such as Google API. This evaluation was completed by recruiting a group of bioinformatics experts and asking them to complete a set of well-defined tasks which helped them assess the performance of our recommender system when considering these relations. Figure 4.1 shows the data-flow diagram for the steps that should be undertaken to reach to each semantic relation that we have discovered.

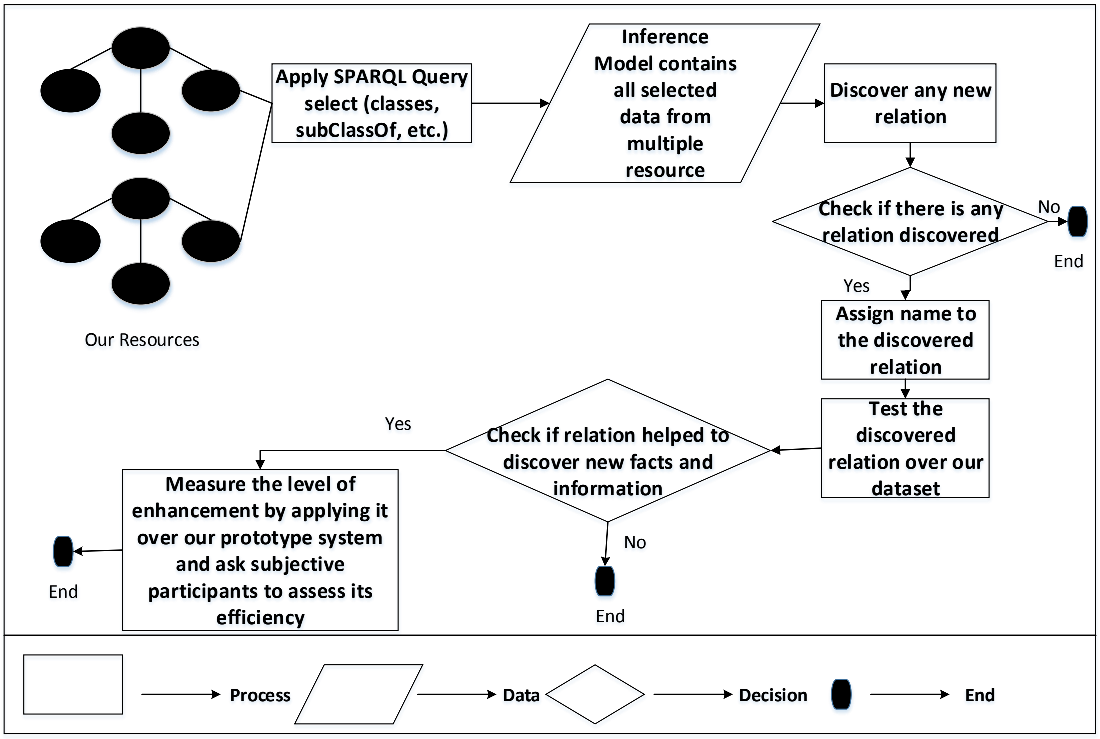

Our Resources
Apply SPARQL Query select (classes, subClassOf, etc.)
Inference Model contains all selected data from multiple resource
Discover any new relation
Check if there is any relation discovered
No
End
Yes
Assign name to the discovered relation
Test the discovered relation over our dataset
Check if relation helped to discover new facts and information
Yes
No
End
Measure the level of enhancement by applying it over our prototype system and ask subjective participants to assess its efficiency
End
Process
Data
Decision
End

Figure 4.1: Data Flow Digram for Semantic Rules Discovering.

4.2.2 Extracting Information from Multiple Resources

As we have mentioned in the previous sections, our approach will deal with different bioinformatics resources (i.e. PO, GO, BLD, ODP ontologies and the Wikipedia corpus) that may have different structures. Thus, to extract information from these resources, they should be in OWL format to be extracted and exploited successfully. Extracting information from OWL files will be done by executing a SPARQL query through the necessary data, since a SPARQL query allows us to determine the specific type of data to be extracted and reasoned instead of extracting all information in each resource. Extracting more than the necessary information may complicate the reasoning process by extracting unwanted triples. These triples may cause some difficulties in the inference process and some technical issues such as heap size or increasing the combination of inferred relations, which contributes to entering the inference process in an infinite loop and causes the machine to run out of memory. Thus, the steps that will be used to extract and reason with data from multiple resources will be described conceptually in this section. This is to show how user profiles will be enriched with additional relations and information to enhance the accuracy of the provided recommendations. Hence, (i) after preparing all of the resources to be in a single format (i.e. OWL), our reasoning method will extract specific types of data. Then, (ii) the method will assign them to an inference model that can be processed by our reasoning method. After that (iii), our reasoning method will apply a set of semantic rules (which will be discussed in detail in the following section) that were designed to infer new relations and information that may exist among the resources. Then, (iv) it will represent the inferred relations and information in a semantic network, which will be connected to each profile based on the similarities between the users' preferences and the semantic network's concepts. Finally, (v) the user preferences and queries will be enriched with information gained from the discovered semantic relation. Figure 4.2 shows the steps that were taken to extract and infer semantic relations between various resources.

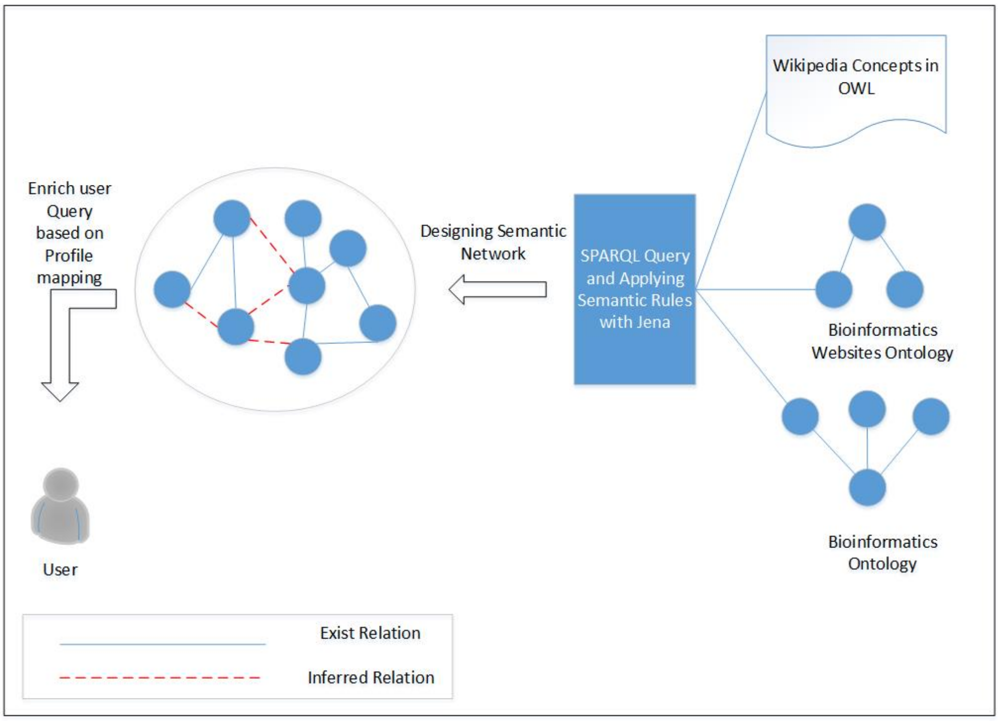

Wikipedia Concepts in OWL
Enrich user Query based on Profile mapping
Designing Semantic Network
SPARQL Query and Applying Semantic Rules with Jena
Bioinformatics Websites Ontology
Bioinformatics Ontology
User
Exist Relation
Inferred Relation

Figure 4.2: Extracting Semantic Relations from different Resources.

4.2.3 Reasoning Rules and Inferences Relations

A semantic web rule can be defined as a conditional statement where an event or set of events will be fired whenever the conditional statement is satisfied [206]. The Semantic Web Rule Language (SWRL) [82] represents the language used to design a semantic rule that can deal with OWL Descriptive Language ontologies. Thus, the idea behind the semantic rule is to apply some assumptions to existing data (i.e. ontology) so that whenever the processed data have these assumptions, the rule will be fired and produce some inference data. The semantic rule can be represented by different semantic rules such as SWRL and Jena. In general, the rule has some components essential to any rule to be fired, namely a body (i.e. a part that contains an if-clause statement and represents the most important part of the rule), the head (i.e. the part that contains a then-clause statement and based on the first part of the semantic rule) and a rule label [206]. Jena rules were considered to represent our approach's rules, since they are more compatible with using the Jena framework. We used Jena rules to be constrained with our model, where rule reasoners can be bounded by a model or schema [206]. The semantic rules can be forward rules, backward rules or hybrid rules that contain both situations. A forward rule fires whenever a new rule is added; however, a backward rule only fires when a query is applied to an associated model [206]. For example, our recommender approach has a set of defined rules that will be applied on the inferred model, which contains some classes that were extracted from multiple resources. Figure 4.3 shows an abstract view of applying semantic rules in the Jena framework.

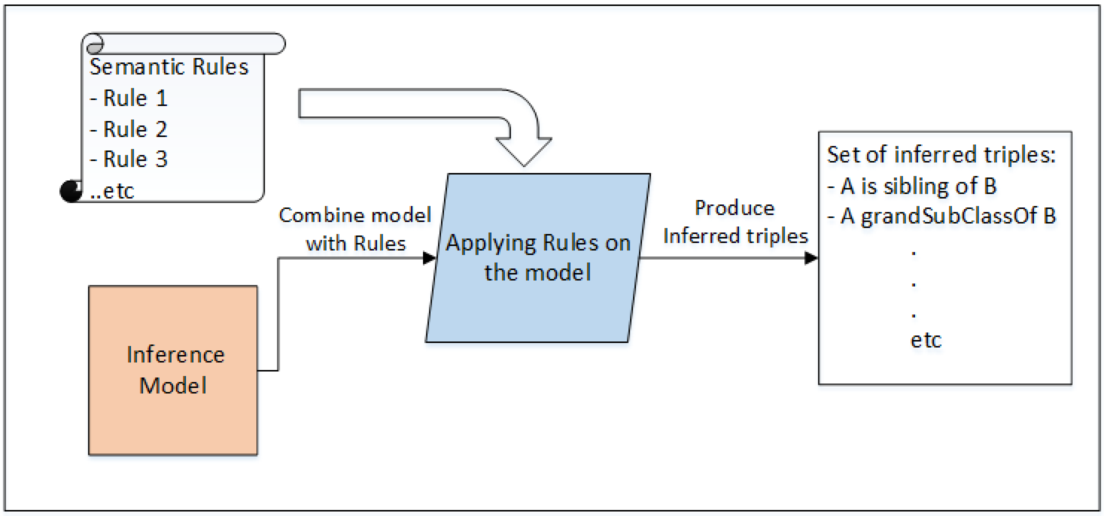


Figure 4.3: Applying Semantic Rules on Inferred Model.

Semantic rules can differ based on the nature of the existing relations in processed data. For instance, our dataset consists of several types of relations that can be inferred and exploited to find more triples that can be used to find rich relations between different classes existing in multiple resources. We have developed seven rules applied in our dataset to extract semantic relations and hidden associations that can be found as a result of information overlapping between multiple bioinformatics resources.

The semantic relations (semantic rules) that we have discovered in this work will be discussed in more details in the following subsection:

4.2.3.1 SuperClassOf Relation

This relation was discovered by following the previous steps in section 4.2.1: in Step 3, we noticed that our dataset already had a relation called “subClassOf” (i.e. the class knows its parent), but it did not have the opposite relation (i.e. the parent knows that it has a child). Thus, we have called this relation a “superClassOf” relation.

The following is an example of the discovered relation: Assume that we have a protein called PR_000000033, which defined as “A protein with a core domain composition consisting of an N-terminal cytosolic domain, a type II transmembrane domain and a C-terminal TNF domain (Pfam:PF00229)”[1]. This is a subClassOf another protein called PR_000000001, which is defined as “An amino acid chain that is produced de novo by ribosome-mediated translation of a genetically-encoded mRNA. [PRO:DAN, PRO:WCB]”[1]. Thus, by following the previous steps, specifically Step 3, we noticed that there is a relation that could be inferred which connects the parent with the child, or in other words, it makes the parent aware of all children that it has. Therefore, we have defined a new relation and called it superClassOf relation as mentioned in Step number 4 in the previous steps. This relation allows the parent (PR_000000001) in this example to be aware of its children has. This relation defined as a semantic rule which fire during the reasoning whenever a subclass relation appears between couple of concepts in our processed dataset. This example represents the Step 5, which involves testing this relation over a real dataset in order to discover its ability to infer new information. This relation is illustrated in Figure 4.4.

[1] http://pir.georgetown.edu/pro/

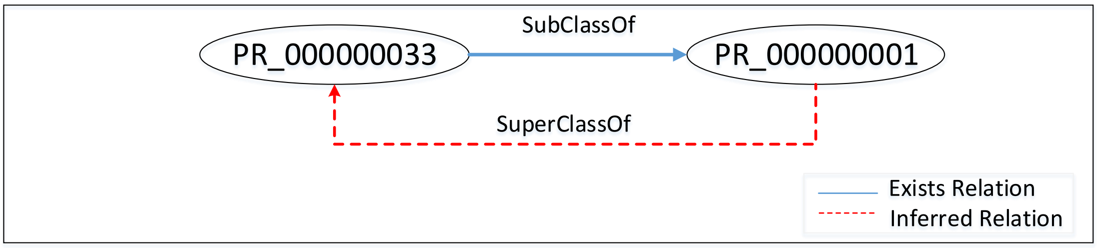


Figure 4.4: SuperClassOf Relation.

4.2.3.2 GrandSubClassOf (transitive) Relation

Similarly to the previous relation, this relation was discovered by following the previous steps in section 4.2.1. Then, in Step 3, we noticed that there was an indirect relation between concepts and their grandparents that could be inferred and exploited to enhance provided services, such as recommendations or searches. In Step 4, we considered the name GrandSubClassOf (or transitive), based on the inferred relation that connected the pair of concepts and because this name reflected the relation that could be found between concepts.

The following is an example of the discovered relation: Assume we have a gene called GO_0048589, which is defined as “The increase in size or mass of an entire organism, a part of an organism or a cell, where the increase in size or mass has the specific outcome of the progression of the organism over time from one condition to another”[1]. This is a subClassOf GO_0001547, which is defined as an “increase in size of antral follicles due to cell proliferation and/or growth of the antral cavity”[1]; and GO_0001547 is a subclass of *Thing*, then the relation that can be inferred from such a relation is a GrandSubClassOf (or transitive) relation between the class “Thing” and GO_0048589. This rule is illustrated in Figure 4.5.

[1] http://geneontology.org/

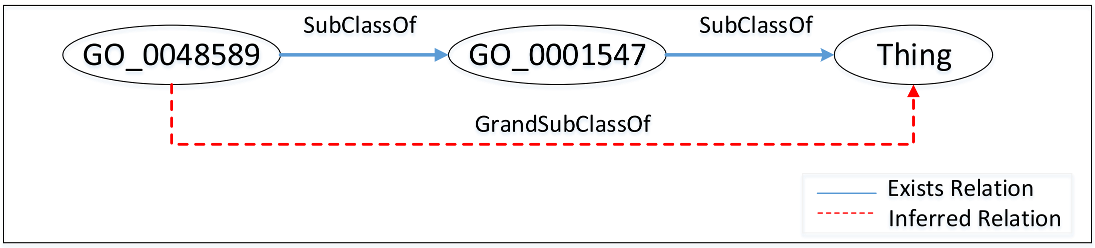


Figure 4.5: Transitive Relation.

4.2.3.3 Sibling Relation

By following the aforementioned steps in section 4.2.1, when we reached Step 3, we observed that the name of the concepts is unique in all of our resources; this is the main condition to satisfy this sibling relation, since different resources may describe a specific concept in a slightly different way and some information may be mentioned in one resource and ignored in another. Thus, Step 3 led to the discovery of a new relation that could be inferred when a concept exists in more than one resource and it has children which mentioned in a resource but not all of them in others or when different children were mentioned in different resources. Also, relation can be inferred between all concepts that have the same parent even if they are in the same resource. Then, in Step 4, we decided to call this relation a "sibling" relation, which is the most appropriate name to reflect the meaning or role of this relation.

The following is an example of the discovered relation. A protein, PR_000002145, which is defined in PO as "a CD14 molecule isoform 1 that has been processed by proteolytic cleavage"[1], inferred that it has a sibling relation with GO_0001404, which is defined in GO as "OBSOLETE Growth of a pathogenic organism that results in penetration into cells or tissues of the host organism. This often (but not necessarily) includes a filamentous growth form, and also can include secretion of proteases and lipases to break down host tissue"[2], where both classes are categorised under a class called "ObsoleteClass" that exists in both

[1] http://pir.georgetown.edu/pro/

[2] http://geneontology.org/

resources, PO and GO. Our approach can exploit such relations to enhance the preciseness of retrieved articles when someone searches an article discussing one of these classes.

Another example of this relation: is as follows. There is a gene called GO_0043234, which is defined as "any macromolecular complex composed of two or more polypeptide subunits, which may or may not be identical. Protein complexes may have other associated non-protein prosthetic groups, such as nucleotides, metal ions or other small molecules"[1]. This gene exists in two of our resources (i.e. PO and GO), and each resource has mentioned a set of children for this gene, but these children were mentioned only in one resource but not in the other. For example, the PO resource mentioned PR_000025402, which is defined as "a protein complex that is a membrane-bound heterodimeric co-receptor for MHC class-I antigen/T-cell receptor interaction. [PMID:18275828,PMID:3264320, PRO:DAN]"[2], as a child of only this gene. However, GO mentioned GO_0000148, defined as "a protein complex that catalyzes the transfer of a glucose group from UDP-glucose to a (1-&gt;3)-beta-D-glucan chain"[1]; GO_ 0008247, defined as "an enzyme complex composed of two catalytic alpha subunits, which form a catalytic dimer, and a non-catalytic, regulatory beta subunit; the catalytic dimer may be an alpha1/alpha1 or alpha2/alpha2 homodimer, or an alpha1/alpha2 heterodimer. Modulates the action of platelet-activating factor (PAF)"[1]; GO_1902508, defined as "a protein complex which is capable of 2-iminoacetate synthase activity"[1]; and GO _0009316, defined as "a heterodimeric enzyme complex composed of sub-units leuC and leuD. Catalyzes the isomerization between 2-isopropylmalate and 3- isopropylmalate, via the formation of 2-isopropylmaleate"[1] as children of GO_0043234. Thus, our method will infer the relation between sets of genes in the GO resource and the protein in the PO resource and refer to it as a "sibling" relation; this relation will be exploited to enhance the accuracy of provided recommendations. This relation is shown in Figure 4.6.

[1] http://geneontology.org/

[2] http://pir.georgetown.edu/pro/

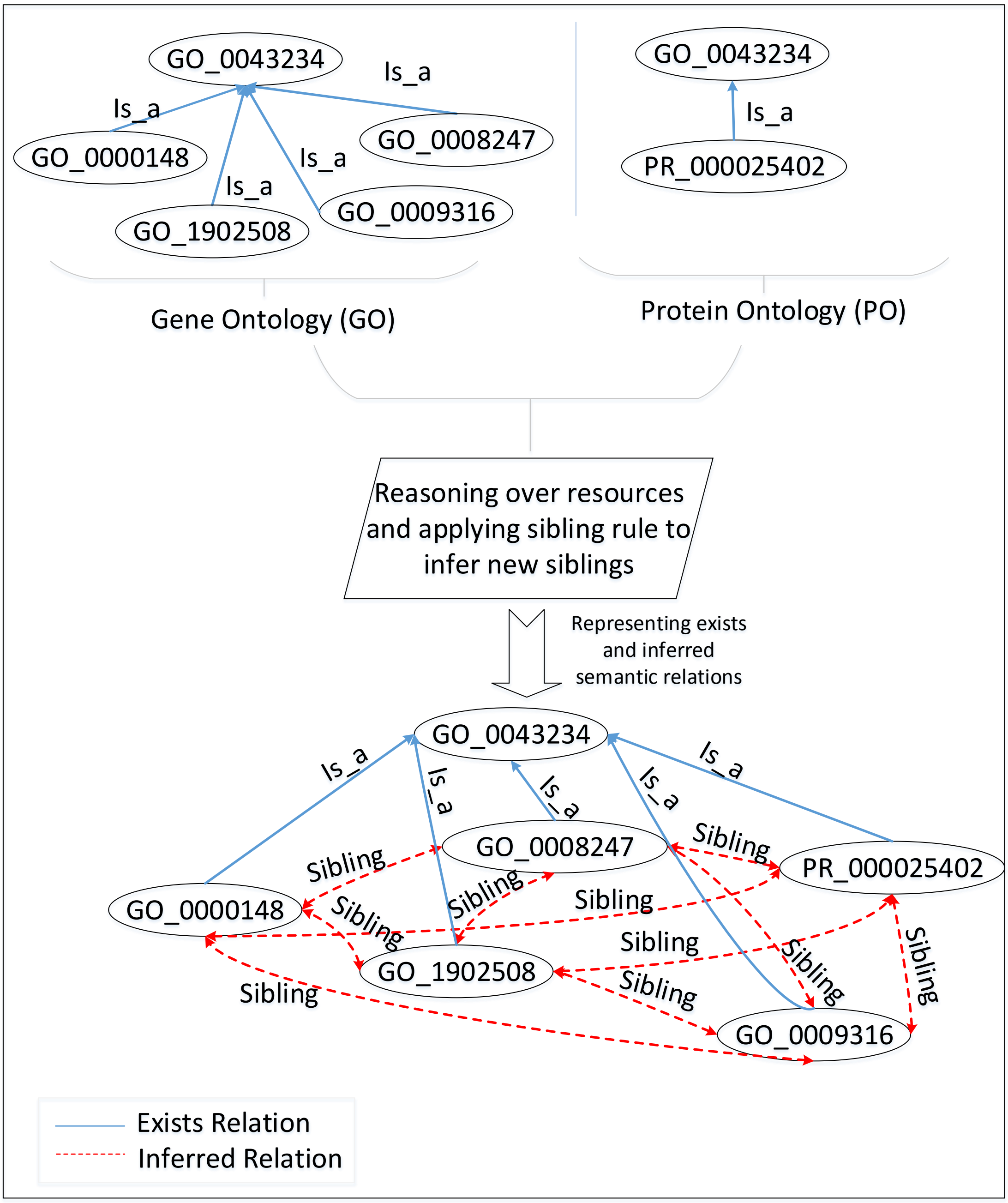


Figure 4.6: Sibling Relation.

4.2.3.4 Is_type_Of Relation

After considering the former steps in section 4.2.1 and completing Step 3, we can determine whether there is a relation that can be inferred between two concepts, especially when the two concepts exist in two different resources, since a resource may mention information that has not been mentioned in the other resource. This led us to infer a new type of relation, where some resources did not include all information about specific concept, such as the type. Then, in Step 4, we decided to call this relation an "Is_type_Of" relation, since this is the most appropriate name that can describe this relation.

The following is an example of the discovered relation. Going back to Example 1.1, there was an inferred relation between GO_0044445, defined as "any constituent part of cytosol, that part of the cytoplasm that does not contain membranous or particulate subcellular components"[1], and PR_000027247, defined as "a cytosolic creatine kinase complex that is a heterodimer of a B-type subunit and an M-type subunit. [PMID:8430764, PRO:DAN]"[2], which occurs as a result of considering multiple resources. Thus, from this relation, our method will infer the "Is_type_Of" relation for GO_0044445, which stems from PR_000027247 through GO_0002186, which is defined as "a dimeric protein complex having creatine kinase activity."[1]. As mentioned in the PO resource, the protein PR_000027247 has a type "complex", so by applying the "Is_type_Of" rule on this protein, GO_0002186 will gain a "complex" type, since it is the direct parent of PR_000027247. After that, GO_0044445 will gain an "Is_type_Of" relation, which is a complex of GO_0002186, because it is GO_ 0002186's direct child. Thus, the "Is_type_Of" relation works for both single and multiple resources. This relation is illustrated in Figure 4.7.

[1] http://geneontology.org/

[2] http://pir.georgetown.edu/pro/

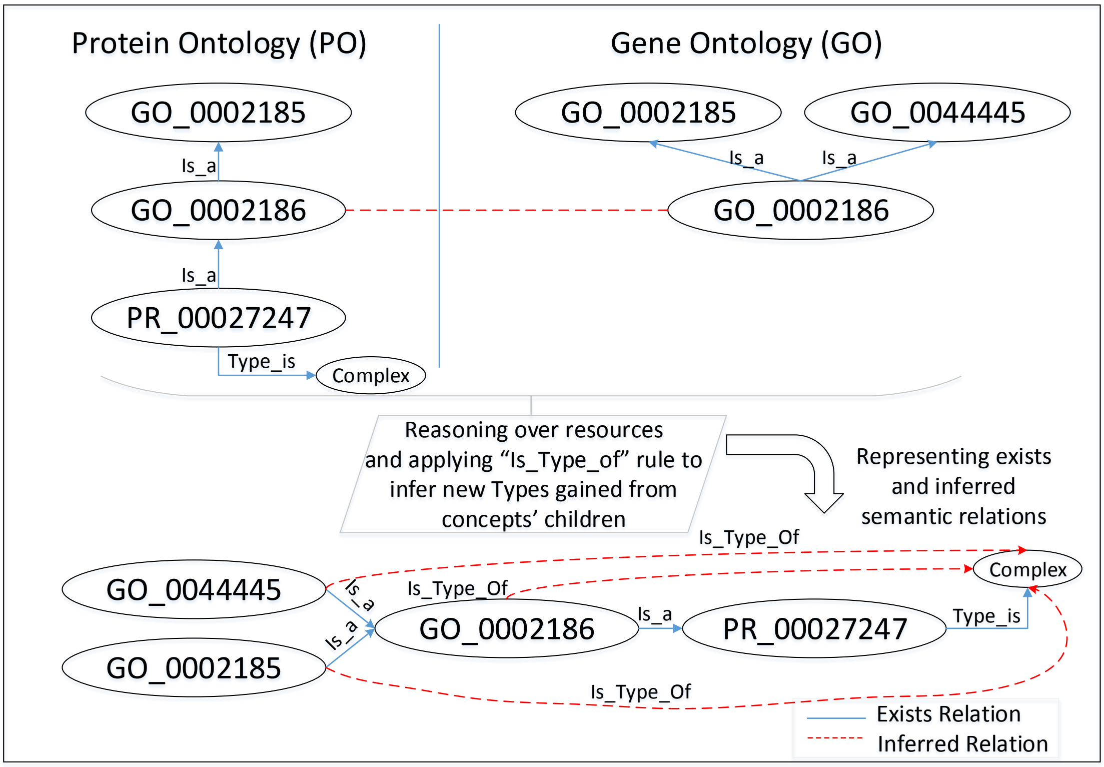


Figure 4.7: Is_Type_Of Relation.

4.2.3.5 SameAs Relation

Following the same steps in section 4.2.1 as aforementioned, in Step 3 we discovered that two concepts can exist in two different resources under different names but have similar descriptions. Thus, we called this relation a "SameAs" relation, since this name may be the most appropriate name to reflect this relation.

The following is an example of the discovered relation: Assume that we have a concept which is mentioned in two of our processed resources. However, these resources give this concept different names, even though their descriptions are similar. For instance, PR_000007595 is called "glutamate carboxypeptidase 2"[1] in the PO resource. However, the same concept is called FOLH1/ GO_REF:0000038

[1] http://pir.georgetown.edu/pro/

is called " glutamate carboxypeptidase 2"[1] in the GO resource. Thus, this relation works for both single and multiple resources. This relation is shown in Figure 4.8.

Another example of this relation is a class called "Zinc" being the same as a class called "Cinc". If "Zinc" exists in two of our resources under different names or labels, such as "Cinc", this rule will compare the description of the two concepts. If the concepts have similar descriptions, they have a "sameAs" relation.

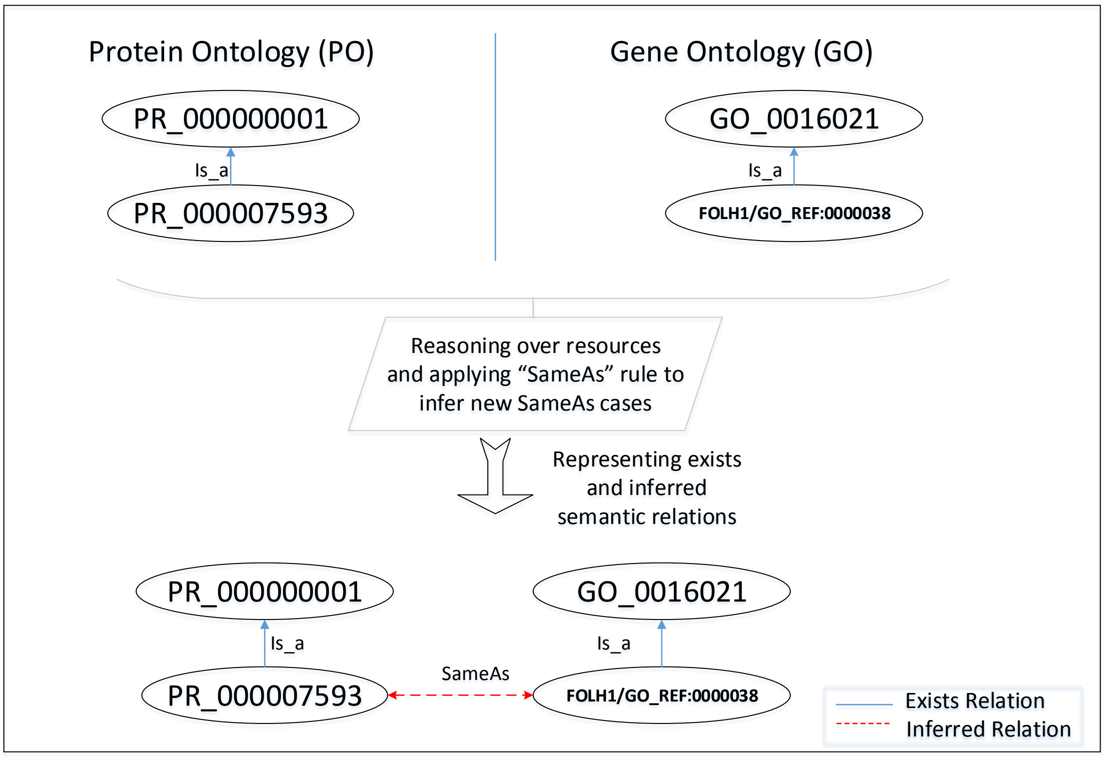


Figure 4.8: SameAs Relation.

4.2.3.6 Equivalent Relation

By following the aforementioned steps in section 4.2.1, specifically Step 3, we can observe that there is a relation that can be inferred when a class consists of class and TransitiveProperty, regardless of whether they appear in the same or different resources. This case could be found when a class is comprised of a restriction

[1] http://geneontology.org/

of class and TransitiveProperty, and these couples may be available in the same resource or even in different resources. Thus, in Step 4, we called this relation an "Equivalent" relation, since it is the most appropriate name for this relation.

An example of the discovered relation follows. Referring to Tom in Example 1.1, if we apply our rule to the concept GO_0002186 which has *TransitiveProperty* which is BFO_0000050 defined as "part of"[1] and shares *someValues* with GO_0005829, which is defined as "the part of the cytoplasm that does not contain organelles but which does contain other particulate matter, such as protein complexes"[1], then our rule will infer that classes GO_0002186 and GO_0005829 are equals. This will lead researchers to discover many facts, especially when using multiple resources, such as PO and GO. This relation will infer that PR_000027247 is a subclassOf these two classes. So, if Tom looks at this information from a single resource (i.e. PO) without considering equivalent relations, then he will able to find that PR_000027247 is a subclassOf GO_0002186 only. But, he will not infer that PR_000027247 is subclass GO_0005829, which will prevent him from discovering new facts and relations gained as a result of applying the equivalent relation over multiple resources. Figure 4.9 illustrates this rule.

[1] http://geneontology.org/

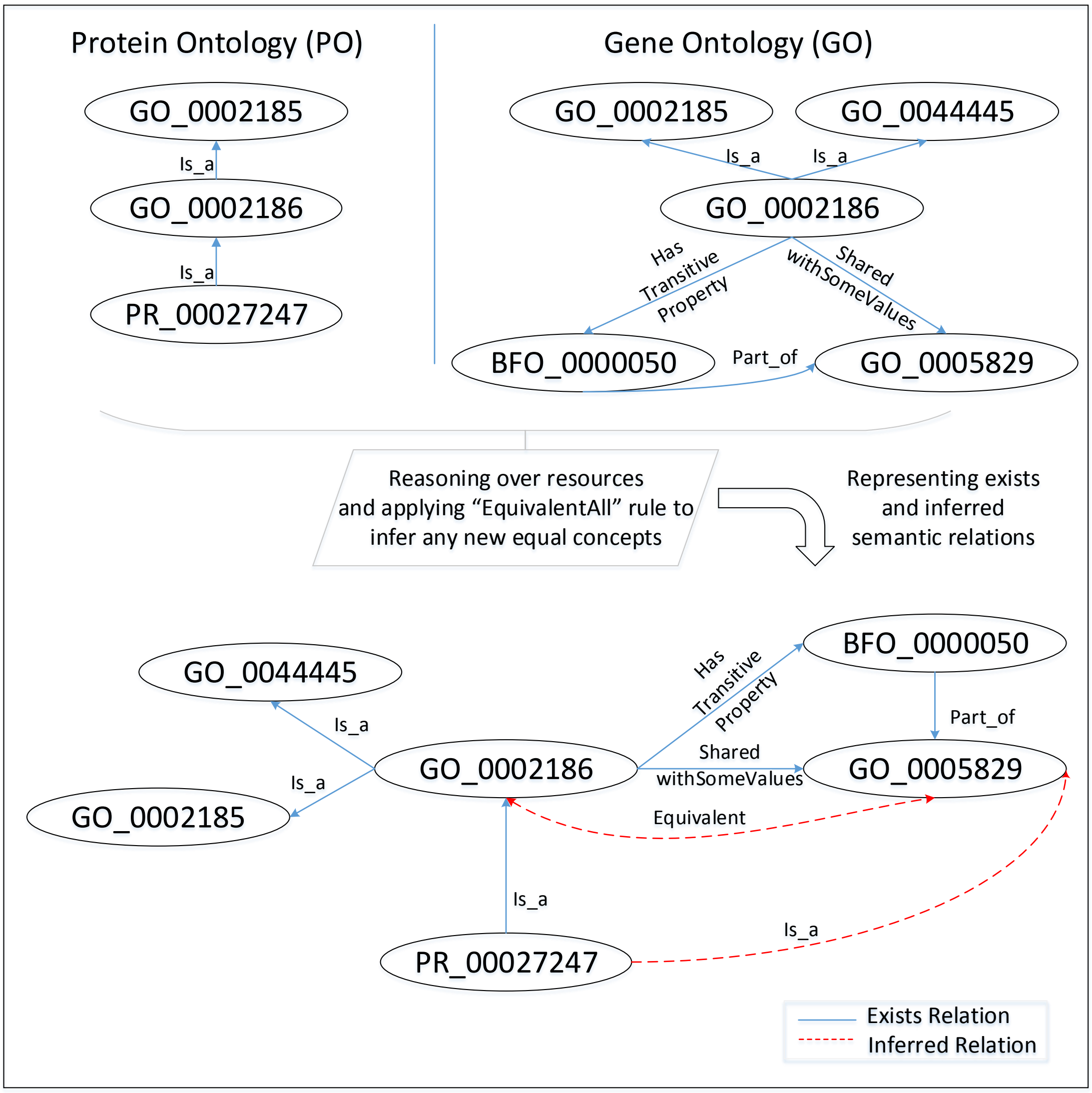


Figure 4.9: Equivalent Relation.

4.2.3.7 Semantic Similarity Relation

By following the previous steps in section 4.2.1, specifically Step 3, we observed that there is an inferable relationship between different concepts that exist in the same or in different resources. This relation could help us discover new information and facts that would not be appear from the direct relation. Thus, in Step

4, we decided to call this relation "semantic similarity" relation, since it represents calculating the semantic similarity between different concepts in our dataset. However, it differs from the other relevant semantic similarity method in its way of calculating or deciding the semantic similarity between two concepts, since this relation is calculated by following a new method of calculating the semantic similarity, which increases the opportunity to find more semantic similarity cases. It calculates the concepts' similarities and then calculates the semantic similarities between the concepts that exceed our suggested threshold during the reasoning process. So, this method takes into account concepts' similarity's score and semantic similarity score to consider any two concepts having semantic similarity. This can allow researchers to find additional similarity cases between the concepts.

The following is an example of the discovered relation. Suppose that a bioinformatician is interested in a concept that exists in gene ontology called GO_0000127, defined as "A heterotrimeric transcription factor complex that is involved in regulating transcription from RNA polymerase III (Pol III) promoters. TFIIIC contains three conserved subunits that associate with the proximal Pol III promoter element, and additional subunits that associate with sequence elements downstream of the promoter and are more diverged among species. It also functions as a boundary element to partition genome content into distinct domains outside Pol III promoter regions"[1]. This relation will help the bioinformatician infer any concept that has semantic similarity with this gene, even if the inferred concept is not located in the same resource. Thus, a concept called PR_000000380, defined as "a transcription factor Sp1 that is a translation product of a mature transcript of the SP1 gene, including all coding exons"[2], which is located in the protein ontology, will be inferred as a concept that has semantic similarity with the preferred concept, even though the two concepts exist in different resources. This relation can handle the semantic similarity between concepts in a single resource or in multiple resources. This example represents Step 5, which involves testing the discovered relation in our dataset to measure its applicability and efficiency to infer new information that would not have appeared if the user searched for

[1] http://geneontology.org/

[2] http://pir.georgetown.edu/pro/

these data in a single resource or without considering this relation. Figure 4.10 illustrates this rule.

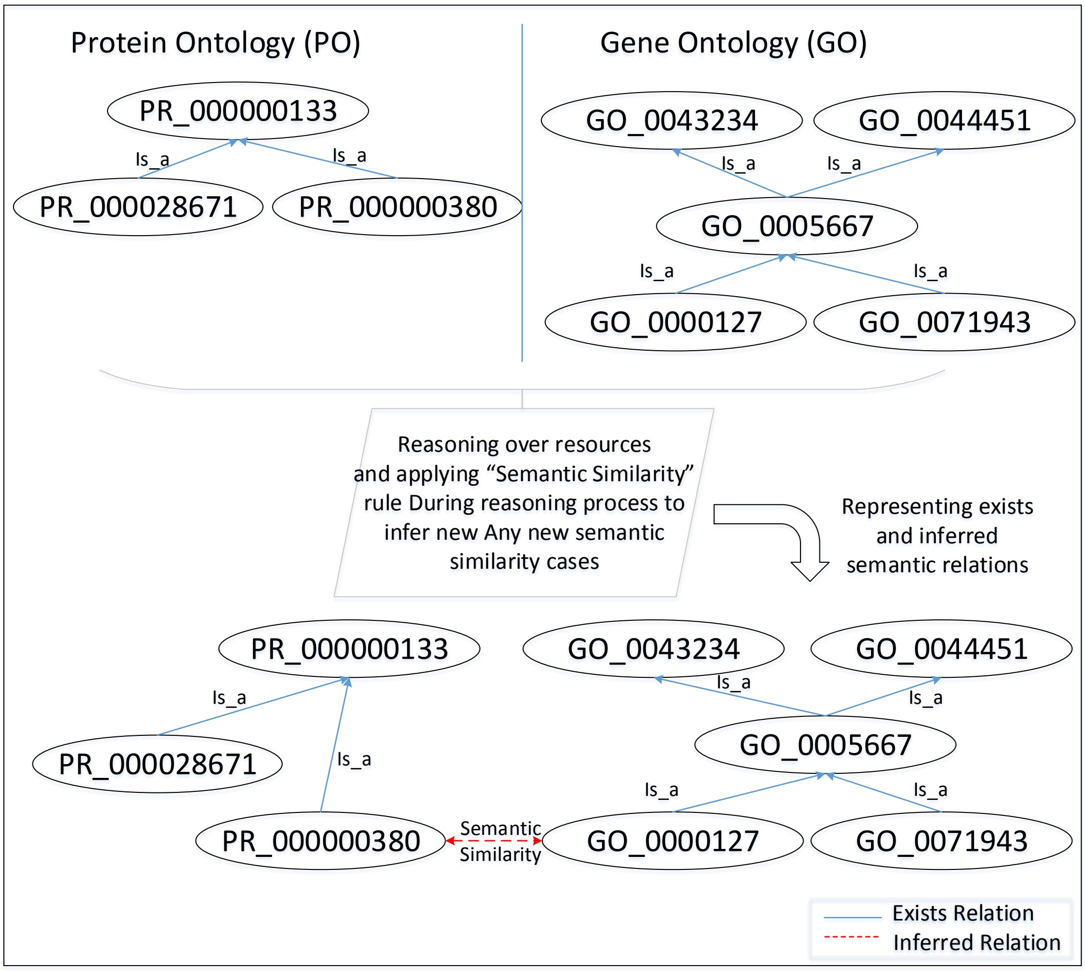


Figure 4.10: Semantic Similarity Relation.

Finally, the strict relationship between the former semantic relations is that all of these relations try to find hidden information and relations that can be exploited by our recommender system to enhance the accuracy of the provided recommendations. Moreover, these relations can help the researcher broaden his/her horizons and discover new facts and information that would not be found if these inferred semantic relations were not considered in the recommendations provided to the researcher.

Furthermore, figure 4.11 shows the different reasoning components and illustrates the general procedure that will be taken in the aforementioned examples to infer a new relation or find extra information gained as a result of information overlapping between different bioinformatics resources.

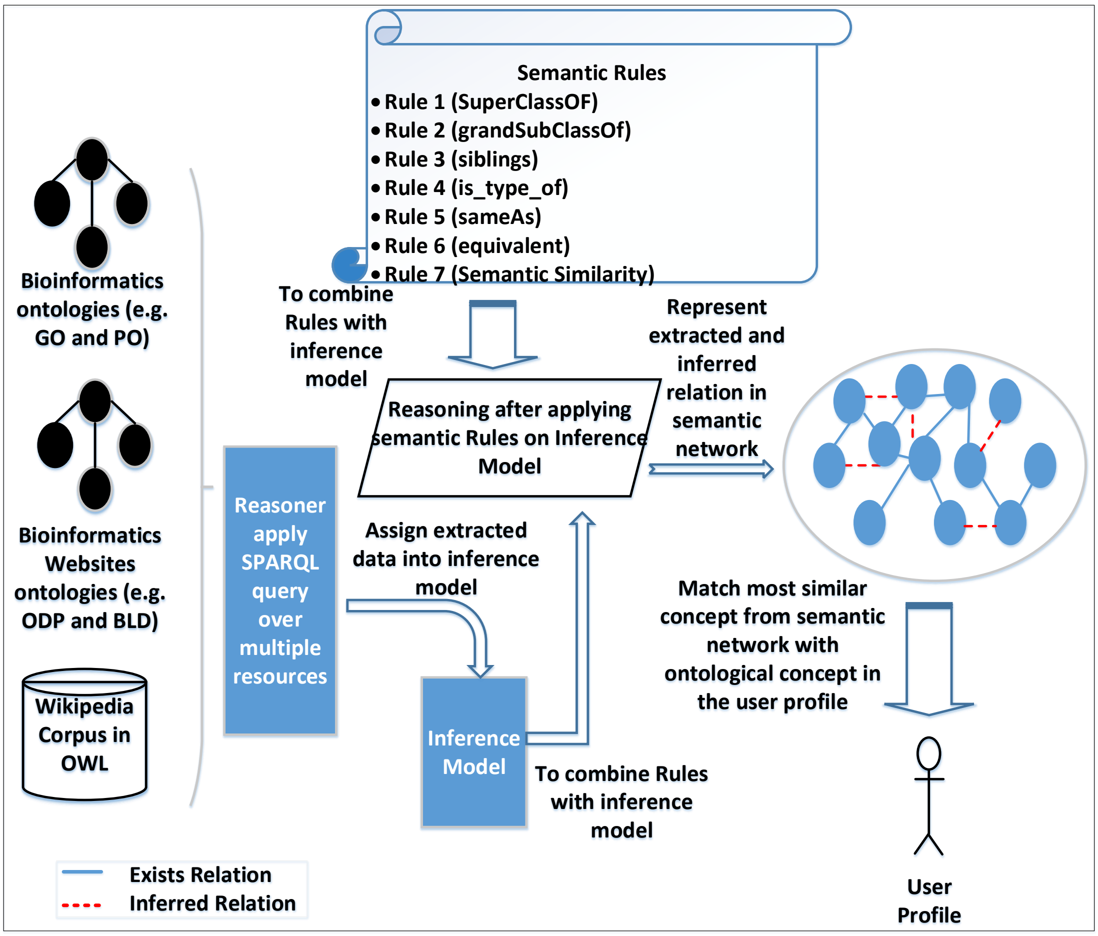

Semantic Rules
• Rule 1 (SuperClassOF)
• Rule 2 (grandSubClassOf)
• Rule 3 (siblings)
• Rule 4 (is_type_of)
• Rule 5 (sameAs)
• Rule 6 (equivalent)
• Rule 7 (Semantic Similarity)
Bioinformatics ontologies (e.g. GO and PO)
Bioinformatics Websites ontologies (e.g. ODP and BLD)
Wikipedia Corpus in OWL
To combine Rules with inference model
Reasoning after applying Semantic Rules on Inference Model
Represent extracted and inferred relation in semantic network
Reasoner apply SPARQL query over multiple resources
Assign extracted data into inference model
Inference Model
To combine Rules with inference model
Match most similar concept from semantic network with ontological concept in the user profile
User Profile
Exists Relation
Inferred Relation

Figure 4.11: Reasoning Components.

Thus, as shown in figure 4.11 our semantic methods are able to handle multiple resources and this task was assigned to the reasoner included in our framework. It selects the required data from each resource and then assigns them into an inference model. After that, the semantic rules we have designed will be combined with the inference model, in which they will support our reasoner method to infer new semantic relations and associations that may exist as a result of information overlapping between multiple resources. Then, the inferred data will be represented as an inferred semantic network, which can be exploited by the user profile to improve the quality of the provided recommendations. Thus, our semantic methods and semantic rules can handle single and multiple resources in OWL format, and can infer new types of relations (such as sibling and semantic similarity) that had not been inferred or used before. These features distinguish our semantic methods and semantic rules from other related methods.

4.3 Inferred Semantic Network

A semantic network can be defined as a set of concepts, nodes or classes connected to each other by semantic links [207]. Concepts in the semantic network can be linked to each other by more than one semantic link. For instance, let us discuss a real example from our dataset and assume that we have a protein called "PR_000000033"[1] which is defined as *"A protein with a core domain composition consisting of an N-terminal cytosolic domain, a type II transmembrane domain and a C-terminal TNF domain (PF00229)"*[1]. It has an exchangeable relation with a protein called "PR_000000001"[1] defined as *"An amino acid chain that is produced de novo by ribosome-mediated translation of a genetically-encoded mRNA."*[1], where PR_000000033 is a subClassOf PR_000000001. From another perspective, our reasoner inferred that PR_000000001 is a superClasssOf PR_000000033. Thus, semantic concepts can be linked by different types of relations such as "is_a", "has_a", "part_of", where each represents a link between two concepts in the semantic network.

In our approach, the inferred semantic network was used to represent semantic relations and hidden associations that were drawn as a result of the reasoning processes, in combination with the seven semantic rules that were discussed previously, and applied to multiple resources as well as information overlap between different resources. Our assumption in constructing the semantic network was

[1] http://pir.georgetown.edu/pro/

that every orphan concept is a direct subClassOf *Thing*. On the other hand, such an orphan concept will be listed as a child of class *Thing*. However, it will be listed under SuperClassOf *Thing* as well, and so *Thing* will be a SuperClassOf the orphan concept. This process will have a direct mapping between concepts to remove duplicate concepts from the semantic network. Then a set of contents reasoned for each class will be added to the processed class or concept, such as *the list subClassOf, the list of transitive subClassOf (i.e. grandSubClassOf), label, comment, list of superClassOf, list of objectOproperty, list of sameAs, list of is_a_type_of, equivalent classes, siblings or semantic similarities.* After that, each class becomes a subject in one or more triples. All inferred relations and associations will be stored in an XML file, to be exploited by the recommender approach. Figure 4.12 demonstrates the inferred semantic network and gives an overview of the discovery and use of semantic relations. Moreover, there is a filter to remove duplicated concepts such as "GO_0043234" that exist in PO and GO, as shown in the figure 4.12. It supposes that any orphan concept is a subClassOf "Thing". If there is no orphan class, it connects the concept in level 1 with subClassOf (is_a) to "Thing". Each concept in our semantic network went through three stages (i.e. determining classes and inferred triples, removing duplicated classes, and checking whether the inferred data already exist), which represent steps that need to be taken to overcome the inconsistencies and challenges due to the various structures and relations between different resources. This represents a brief description of how our semantic method addresses some inconsistencies between resources. Also, it shows sibling and semantic similarity relations.

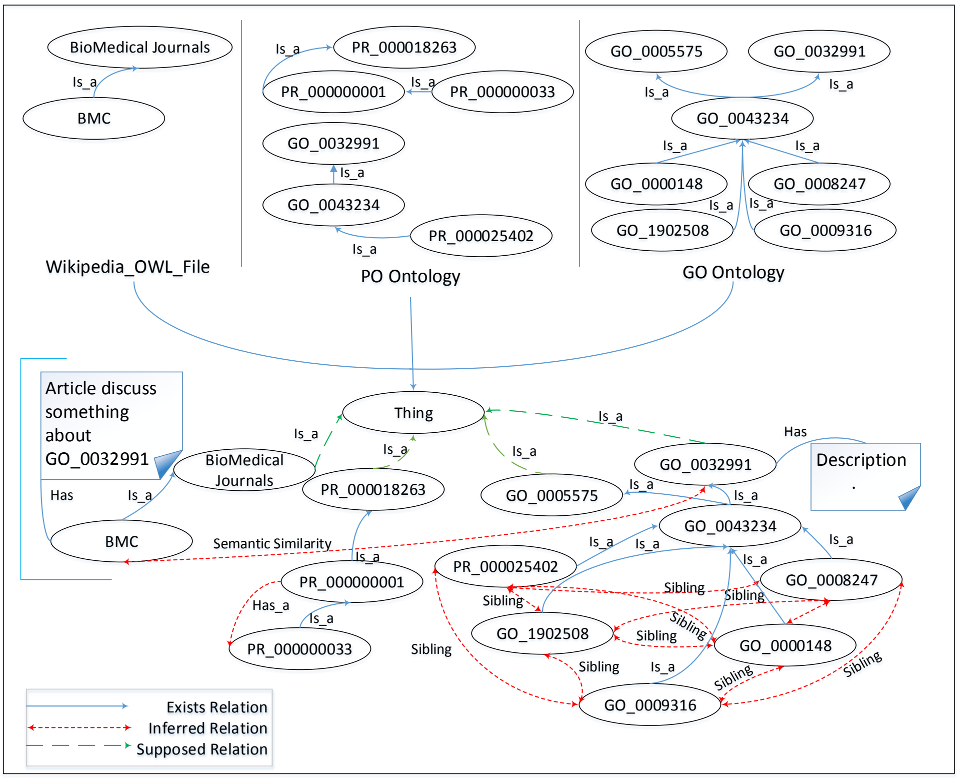


Figure 4.12: Inferred Semantic Network.

4.4 The Semantic Similarity Inference Method

Considering that multiple bioinformatics resources may lead to the discovery of implicit information and relations that need to be extracted, represented and exploited. Therefore, a method needs to be designed to undertake these tasks and enhance the quality of the provided recommendations. There is a range of semantic relations that can be captured as a result of information overlapping between resources that can help enrich queries and extract more accurate content. This was the motivation for designing solutions that can extract and exploit such information. These solutions were embodied in two types of relations, namely sibling relation and semantic similarity inference relation. The former focuses on identifying all concepts that have the same parents with the processed concept even though they are not present in the same resource (section 4.2.3 provided all details about this relation). The latter is concerned with deciding what concepts are semantically similar by considering the semantic similarity and concepts' descriptions similarity between the concepts, and this section focuses on exploiting the semantic similarity relation.

To decide the semantic similarity relation between two concepts, their descriptions should pass three stages: i) After the reasoning process is started through the selected information from multiple resources that were aggregated in an inference model, both concepts should have comments or descriptions to be processed by our semantic similarity method. Thus, if this condition is satisfied, then the custom built-in rule 4.8 will be fired, which calls the semantic similarity method to process the selected elements. ii) Equation 4.1 was adapted from [49] will be used to calculate the similarity between the descriptions of the two concepts, and the similarity score should exceed our suggested threshold (this will be discussed in more detail in the following chapter) in order to pass to the final stage.

$$Concepts\,Similarity(D_1, D_2) = \frac{2 * Number\,of\,Similar\,Words\,in\,(D_1, D_2)}{Length\,of\,D_1 + Length\,of\,D_2} \tag{4.1}$$

where D_1 represents a description of the first concept and D_2 also represents a description of the second. This equation is used for calculating the concept similarity between two descriptions of concepts and we call it *ConceptSimilarity*.

Moreover, while the inferred semantic network is being formulated, the semantic similarity between the two concepts should be determined, which has been used by different works [208]. It can be done by calculating the distance between the two concepts through equation 4.4; then, the distance score will be used to calculate the semantic similarity between the two concepts by using equation 4.5. So, in order to calculate the distance, there are two equations that need to be calculated first. The first one is to calculate the weight of two nodes, equation 4.2:

$$W(x, y) = \frac{1}{2^{level(y)}} \tag{4.2}$$

where x represents a concept or node in the semantic network and y also represents a concept in the semantic network.

The second one is to calculate the semantic distance between two nodes, equation 4.3:

$$Sem\_Dist(x, y) = \sum_{z \in shortestPath(x,y)} W_z(x, y) \tag{4.3}$$

where z represents path between x and y.

$$Distance(x, y) = \; minSem\_Dist(x, Thing) + minSem\_Dist(y, Thing) \tag{4.4}$$

“Thing” is the suggested root for all concepts included in the semantic network. “minSem_Dist” represents the shortest path between two concepts in the semantic network.

$$Semantic\, similarity(x, y) = \frac{1}{Distance(x, y) + 1} \tag{4.5}$$

Finally, iii) includes crossing the two scores of equations 4.1 and 4.5 (concepts' similarity and semantic similarity scores), so when their result exceeds our considered threshold, the concepts will be considered to have a semantic similarity relation.

The semantic similarity method or relation is a novel relation that has been extracted as a result of information overlapping among multiple bioinformatics resources. This method can be distinguished by its ability to calculate semantic similarity while the reasoning process is running, which may lead to new semantically similar concepts being inferred that would not appear using the regular method.

4.5 Automatic Method for Updating Semantic Concepts

There is a need to update the underlying ontologies being used. With the availability of computational power, different disciplines have created increasing research and knowledge has expanded, leading to several resources that need to be developed and enhanced in a short period of time. Therefore, we created a method responsible for updating some resources in our dataset. This method involves providing updated information gained from multiple bioinformatics resources, which can enhance the accuracy of the content recommendations (i.e. articles). The component created in our approach will be fully automated by updating our inferred semantic network in some of our semantic resources if they made any update, namely the PO and GO ontologies, since these resources have been prepared to be in the form of OWL. This will allow the update component to contact these resources' websites and download new versions of their ontologies to be compared with the versions that we have; so in case any update is addressed, it will be updated in our resources. Moreover, this method contributes to preserving the former resources updated to ensure that all items recommended to the specialist that are recommended based on these resources (PO and GO) will contain up-to-date information.

Some related works are based on the Ontology Update Language (OUL) [209]. OUL represents a database trigger and SPARQL statement used by many approaches to update their ontologies. For example, Sangers et al. [210] provided a framework based on the RDF model and a SPARQL statement used to update ontologies. However, our component can only handle the OWL models. Moreover, OUL has some drawbacks, such as the inability to support namespaces and only capturing the first match concept, while ignoring the other details. Thus, these disadvantages may make the updating process inaccurate and create many mistakes. As any other work, our component has some shortcomings, in that it cannot update all of the other resources we have, such as ODP, BLD, and Wikipedia terms, since these three resources require developer intervention to be in OWL format. This represents a difficulty facing our component in dealing with these resources to update processes automatically.

4.6 Mapping Semantic Concepts to User Profiles

Exploiting information gained as a result of information overlapping between different resources is essential for this research. This exploitation will ensure that each user of our recommender approach has a rich user profile, which will enhance the precision of the recommended articles. Moreover, it will connect each preference in the user profile with the most similar concept from the inferred semantic network. The connection will be based on the exploited relation, a sibling or semantic similarity method. In other words, if a user profile is enriched with sibling relations, then its preferences will be enriched with the first concept that has a sibling relation (first sibling) with the most similar concept in the inferred semantic network to the concept stored in the user profile. But if the user profile is enriched with a semantic similarity method, then its preferences will be enriched with the concept that satisfies the highest semantic similarity score with the most similar concept in the inferred semantic network to the preference in the user profile. Thus, each user profile will have its own semantic network based on the user's preferred topics or interests, which may change daily, and so a sub-semantic network is designed to be a dynamic network tailored to each user based on his/her preferences stored in his/her profile. Any change in the user profile will change the sub-semantic network. This will ensure that each user will have a profile rich with data gained from the information overlapping among multiple bioinformatics resources. Moreover, this will lead to an updated user profile that will keep the specialist aware of all updates that may happen in a specific domain, such as bioinformatics.

The enrichment performed between the user profile and semantic network has several purposes. It can be exploited to enhance the preciseness of the recommended and retrieved articles. It can also be exploited for re-ranking purposes by organising the retrieved articles based on their priority determined by the bioinformatician. It can be used to enrich the bioinformatician query to return greater accuracy. For instance, let us assume that someone wants to find out about Java as a programming language. This concept has been matched with bioinformatics in the inferred semantic network, and so a specialist's query will be fed with the concept bioinformatics. Thus, the recommended articles will concentrate on the articles that discuss or contain topics about Java; however, the Java articles focussing on bioinformatics such as BioJava and other topics may cover both keywords.

4.7 Conclusions

To sum up, this chapter discussed the main contributions and new methods suggested in this thesis from a conceptual perspective. It showed their importance and how they will support a researcher in the specialist field when such methods and semantic techniques are applied into our recommender approach. They will enrich the specialist's knowledge with recommendations and information gained as a result of information overlapping among multiple bioinformatics resources, and they will help him/her to save time and effort in finding what he/she wants. Furthermore, our recommender approach can provide more efficient results for the specialist search, as the more information that can be utilised on the recommendations, the more accurate results can be gained.

Moreover, the seven rules 4.2-4.8, represent an essential part of the extracting and reasoning processes. This work is the first approach created to extract information from multiple bioinformatics resources using these pre-defined rules (i.e. the six rules (from 4.2 to 4.4), specifically the sibling rule 4.4, which has been tested in this approach) and a custom built-in rule (i.e. semantic similarity 4.8), in order to enhance the recommendations in the specialist search by exploiting information gained from sibling or semantic similarity relations in multiple bioinformatics resources. Furthermore, this approach is also novel in its method of applying these inferred semantic relations to enhance the accuracy of the recommendations provided.

Relevant works by Elenius et al. [81] and Rakib et al. [85], which were discussed in the literature review, used semantic rules called SWRL, but the researchers did not design or use the same rules that we have used. They also did not use our unique method of employing these rules to extract and reason through multiple resources. They used SWRL rules, which are limited in handling complex relations, unlike Jena rules, which are able to handle complex relations. In terms of calculating semantic similarity, a relevant work by Teng et al. [62], which was also discussed in the literature review, introduced a method that calculates functional similarities between GO information content (IC). However, they do not use our new way of calculating semantic similarity between different concepts during the reasoning process that takes into account both the concepts' descriptions similarity and semantic similarity between concepts. Moreover, their work did not apply any inference method when calculating the similarity, which may help to infer new similarity cases that would not appear without considering an inference method and semantic rules, as our semantic similarity method table 4.8 does. Furthermore, all of the inferred data of these relations will be stored in our inferred semantic network, to be exploited by our approach in order to enhance the accuracy of the provided recommendations. This was discussed in detail in section 4.3.

The next chapter will demonstrate our prototype recommender system and how all of the methods discussed are integrated into this prototype system to enhance the precision of the provided recommendations. Moreover, it will include all of the design steps that were considered in each method to be able to work with the other designed methods and techniques.

Chapter 5

Prototype Recommender System

5.1 Introduction

This chapter demonstrates the implementation of all of the aforementioned contributions. Moreover, it discusses the techniques and methods that have been considered in this research to reach its main goal. It discusses the ways that our approach followed to allow these techniques and methods to work together to satisfy the desired goal. This research was intended to support specialists in specific domains such as bioinformatics by providing them with the most relevant content tailored to each user individually based on his/her preference. This will reduce the time and effort that could be consumed to search about a specific topic or information. In addition, it illustrates the tools that were considered and the types of resources that were processed to give a comprehensive view of the process of formulating our recommender approach.

Going further, back to figure 3.1 in Chapter 3, which showed different components of the framework in this work, this chapter will discuss the implementation of the aforementioned methods and will show how they complement each other. i) Firstly, the resources will be prepared to be in mono format (i.e., OWL). Moreover, the system tools and basic algorithms for the recommendations process will be discussed, such as file indexing and search algorithm. Then, ii) the user profile in both experiments 6.3 and 6.4 will be implemented, which includes data collection and mapping a user profile to the semantic network. After that, iii) extracting and reasoning though multiple resources by applying our reasoning method will be performed, which will be supported by semantic rules in order to extract and discover new information and relations that occurred as a result of information overlapping between multiple resources. Then, iv) the semantic network will be implemented, which represents the inferred semantic relations and hidden associations and defeats all the challenges of inconsistencies between different resources' structures. Next, will be v) the implementation of the semantic similarity method needed for programming intervention, which works side-by-side with the semantic similarity rule in order to reach the intended goal. After that, is vi) the implementation of the method designed to keep some of the resources in the inferred semantic network up-to-date; these resources should be in an OWL format, such as GO and PO. Finally, vii) the implementation of the content-based recommender method will be performed, which represents the last target to be completed in the process of our framework to enhance the accuracy of the provided recommendations, which represents the last target to be completed in the process of exploiting semantic relations and associations to enhance the accuracy of the provided recommendations.

5.2 Resources Extractions Implementation and Prototype System Tools

5.2.1 Information Extraction Algorithm from Wikipedia

An extraction algorithm was designed in order to extract information from DBpedia. However, there are several pre-processes that should be undertaken before applying it. The DBpedia Framework [1] needs to be installed, and it is very important for it to be installed in a local machine in order to be used as middleware or API for extraction processes[2]. After that, the next step involves downloading the Wikipedia dump files or MediaWiki so that these will allow the user to retrieve the data from DBpedia. Now, the environment becomes ready and algorithm 1 shows the steps to extract information from the DBpedia. This algorithm was

[1] http://wiki.dbpedia.org/Documentation

[2] There are some bugs that will appear after the installation due to some missing libraries.

designed to be able to extract the data from the DBpedia and prepare it as triples (Subject-Predict-Object). Then, the extracted triples will be formulated in OWL format to be extracted and reasoned by our reasoning method.

Algorithm 1: Extracting Information from Wikipedia

```
Data: List of Terms
Result: an OWL file in form of ontology contains all valuable information about each term such as
        descriptions and relations
Read_and_Normalise(); //Read each term
//Loop through all terms to extract data from DBpedia
for each term in Term_list do
    Create SPARQL Query(term);
    SELECT ?s ?p ?o //This shows variables of subject, predicate and object at the end result
    WHERE
    ?s ?p ?o . //Graph pattern that bind ?s to the subjects of the triple in DBpedia, ?p to the predicates
    and ?o to the objects
    <http://dbpedia.org/resource/one_of_our_terms>?p ?o . // This binds ?p and ?o to the values
    found in triples matched with “one_of_our_terms”
    FILTER(str(?s) = “http://dbpedia.org/resource/one_of_our_terms”) // This filter is to ensure that
    only the triples with the object equal your term end up in the result set
    Connect_to_DBPedia_API(SPARQL Query) // This Function is sending a query to DBPedia
    in order to retrieve information about the term
end
Result2 = Extract_SPARQL_Result (Result1); // This function is to get content of sparql results.
Result3= Save_all_data_in_XML_files(Result2); // All files will be assembled into single file in the next
step.
Result4 = Validate(Result3); // This function is to validate that there is no empty return result.
Result5 =Convert_files_format_to_RDF/XML(Result4); //This function is to save all retrieved data in
a single RDF/XML format file to be ready to be converted to OWL
```

5.2.2 Platforms and Tools

This section outlines some of the tools used to conduct this project and the rationale behind choosing these tools. So, this will provide a full idea about different bioinformatics resources that have been considered in this project and also shows the programming languages and software that have been considered for this purpose:

- **Database:** MySQL 5.0[1] was chosen to represent our database for several reasons:
    - It is free source and can serve different types of applications.
    - It contains different functionalities that contribute to managing data easily and effectively.

[1] http://dev.mysql.com/downloads/mysql/5.0.html

  - I have personal experience in dealing with this tool.

- **Programming Language:** Java was chosen to implement the methods and algorithms in order to construct the prototype recommender system. The reasons behind this selection, are as follows:
  - Java is a multi-platform and portable language [211].
  - It is a fully object-oriented programming language [211].
  - It is suitable for different application sizes (i.e. small, medium and large).
  - I developed several projects using this language as well.

- **Reasoning Tools:**
  - Protégé beta_4.2[1] was chosen as the reasoning tool discover and explore different ontologies of our dataset. There are some characteristics that distinguish this tool from similar ones:
    - This tool is equipped with several reasoners, such as RaserPro[2], Pellet[3], FaCT++[4] and HermiT 1.3.6[5].
    - It is able to handle files with different formats, such as OWL.
    - It provides several services, including comparing two ontologies, merging, extracting classes, sub-classes, graphical view for ontologies and many other services.
  - Jena Framework [212]: A well-known framework that includes libraries in Java. Jena is able to handle files from different formats such as OWL and RDF. Also, it can work with SPARQL queries [72] in order to retrieve or reason through specific information that exists in the processed file.

[1] http://protege.stanford.edu/download/registered.html
[2] http://www.racer-systems.com/
[3] http://clarkparsia.com/pellet/
[4] http://code.google.com/p/factplusplus/
[5] http://hermit-reasoner.com/

- **Files Format:** There are three types of files that will be considered in this project:
  - Extensible Mark-up Language (XML) [213] represents the standard format for different files that we intend to process; this format will be used in the programming and database.
  - RDF describes the grammar for different ontologies with which we will deal.
  - OWL represents the schema for the RDF file, and it will be used in Protégé and Jena for reasoning purposes.
- **Resources (Datasets):** This project deals with different types of resources, such as corpora and ontologies, and these resources are listed below:
  - BMC corpus: This corpus contains bioinformatics content, such as articles.
  - Wikipedia corpus: This corpus contains general content and a subsection for bioinformatics as well as semantic relations that can be exploited to enhance recommendations.
  - ODP ontology: This ontology contains concepts in different disciplines and a website directory for these concepts; we will consider bioinformatics concepts and their websites links of this ontology; thus we will not be concerned with other branches of this ontology.
  - BLD ontology: This follows the same idea of ODP. However, this ontology was designed specifically for bioinformatics concepts, and it contains a website directory for different bioinformatics resources such databases, tools and articles.
  - GO ontology: This contains information about genes and their classifications. It can be used to enrich and increase the level of accuracy of the provided recommendations due to the information overlap between this source and the others.
  - PO ontology: This is an ontology which has information about proteins, their classifications and relations.

5.2.3 Underlying Search Engine Algorithms

Algorithms were designed to handle BMC documents and user queries, since, our prototype system provides articles from the BMC to the specialist as recommendations when he/she submits a query to the system. The first algorithm is the Indexing Pre-Processing algorithm, and it is used to prepare the documents for indexing. Having an index for our dataset will dramatically decrease the time to retrieve results from the BMC corpus when user-submitted keyword(s) are searched. Algorithm 2 shows the Indexing Pre-Processing:

Algorithm 2: Indexing Pre-Processing

```
Data: BMC corpus documents
Result: Indexed files
for each document in BMC corpus do
    //This function is to read document
    Result1= filereader.Read(document);
    //This function changes content to lower case
    Result2 = To_lower_Case(Result1);
    //This function removes stop words from document
    Result3 = removeStopWords(Result2);
    //This function returns words to their stem
    Result4 = Stemmer(Result3);
    //This function preforms indexing for document
    IndexFunction(Result4);
end
```

Algorithm 3 represents a user query and steps undertaken to return results for the submitted query. It returns the most related documents for the submitted query. An inner function has been designed to calculate the time taken to retrieve any document from the BMC corpus. Moreover, in this algorithm, user queries go through the same pipeline as algorithm 2.

Algorithm 3: Searching Process

```
Data: Query
Result: Top 100 ranked documents
// Read user's query
Read(Query);
//This function changes query to lower case
Result1 = To_lower_Case(Query);
//This function removes stop words from query
Result2 = removeStopWords(Result1);
//This function returns query's words to their stem
Result3 = Stemmer(Result2);
//This function is to search for user's query
SearchFunction(Result3);
```

5.3 User Profile Implementation

This section will discuss the ways that we have used to implement the user profiles for each experiment (6.3 and 6.4). This will be discussed in more detail in the following subsections.

5.3.1 User Profile's Implementation for Sibling Experiment

This section will discuss implementing the user profile for the experiment, which assesses the effectiveness of applying our developed sibling method 6.3.

5.3.1.1 User's Data Collection and Browser's Plug-In

There are different ways to collect a user's data implicitly, where each type of data requires specific procedures to be obtained. As mentioned in the previous sections, the data which we intend to collect from the user are diverse and need different techniques to be collected. Firstly, for collecting the surfed URL, there is a tool called *ManicTime*[1], which can be installed on the user's machine. It then automatically collects all surfed URLs, dates and time spent on each website. However, this tool needs other software or tools to support it, due to its inability to collect all of the user's information that is needed such as bookmarked URLs and number of clicked links. For this reason, a browser plug-in has been designed which can work together with *ManicTime* in order to collect all required information.

The plug-in has been designed to undertake two tasks: saving bookmarked URLs and the number of clicked links at a website visited by a user. This will be done in two types of browsers, namely, Google Chrome[2] and Mozilla Firefox[3]. The plug-in is connected to the MySQL database to store the collected data, and it runs through Apache Tomcat 7.0.42[4]. The Tomcat server has been set up to start and run automatically in the user's machine in order to allow the plug-in to perform its task without any disturbance to or intervention from the user. So, when the user opens one of the mentioned browsers, a small window will pop up

[1] http://www.manictime.com/

[2] https://www.google.com/intl/en/chrome/browser/

[3] http://www.mozilla.org/en-US/firefox/new/

[4] http://tomcat.apache.org/

with the message *“Info: Websocket connection opened”* (appendix B) this message reflects that the plug-in started working successfully. In case the plug-in does not work successfully, the following message will appear: *“Info: Websocket connection closed”*. Then, after the plug-in starts, any bookmarked page or any link clicked inside a webpage will be stored in the database.

5.3.1.2 Adding and Updating Mechanism

There is a component that is responsible for adding interests to the user profile, and it performs four main steps. Firstly, the *ManicTime* tool works automatically when the browsing process starts and collects the URLs, time spent in each URL and the date that URL was accessed. A plug-in that is connected to the browser (i.e. works side by side with *ManicTime*) collects interactions that *ManicTime* is unable to collect such as bookmarks and number of clicks on each website. Secondly, all these data are collated, and the interests are added by applying the following equation:

$$User's\,Term\,Frequency = \frac{Simi_i + Frq_i + Nv_i + T_i + B_i + C_i}{Simi_i + Frq_i + Nv_i + T_i + B_i + C_i + a} \tag{5.1}$$

$Simi_i$ represents the cosine similarity score that the surfed URL satisfies with the ontology concept. Frq_i represents the frequency of URL that has some similarity to the ontology concept. Nv_i represents the number of visits to the URL. T_i represents the total time spent reading the concept. B_i represents whether the webpage is bookmarked or not; this takes a value of 1 when the page has been bookmarked by the user and 0 if not. C_i represents the number of clicks in this website; a is a constant that equals 100, as this number is the best value we have reached to make the $User's\,Term\,Frequency$ between 0 and 1. The results are normalised by dividing each result by the numerator value plus 100. Finally, i represents the preferences stored in the user profile. In the third step, after calculating the term frequency for each term, all items with frequency values above 0.1 (i.e. the thresholds that were identified based on several runs, and this threshold provided the best recommendations) are stored with recent visits, which reflect the last time that the user visited a specific URL; this information can be important in specific situations. For instance, let us assume that a user has the same term frequency score for two different concepts, but the times of recent visits for the two concepts are different. So, in such a case, the times of the recent visit will reflect the importance of each concept in comparison with the other to determine which concept is preferred by the user. In the fourth step, update and delete mechanisms are applied to generate updated user profiles that better represent the user preferences. So, the update mechanism will increase term frequency (term weight) for each visit by 0.05 (i.e. the thresholds were identified based on several runs, since this amount increases to fit with the number of maximum days since the last visit, which is 20 days), which represents the amount of daily increase and decrease. Therefore, when the website is visited by the user, it will increase and the opposite will take place otherwise. Moreover, the delete mechanism will run when the user's visit length is less than a threshold of 10 seconds, the number of days since the last visit is more than 20 or the term frequency is less than 0.1. The deleting or forgetting mechanism will be discussed in more detail in the next section, where all reasons behind considering these thresholds for deleting preferences will be clarified. Algorithm 4 describes in more detail the managing of the user profile (add, update and delete functionalities).

Algorithm 4: Managing Concepts in the User Profile

```
Data: User's ID, URL_OPD_Similarity, Term_Frequency, Num_of_Visit, Total_time_Spent, Num_of_clicks,
      Bookmark and Date_of_lastVisit
Result: List of preferences
//Adding Preferences to the user profile
for counter <number_of_matched_URL_and_ODP do
    perform equation (5.1);
    add_preference(); //This step to add user preference and matched with most similar concept to
    exploit sibling or semantic similarity relation
    counter++;
end
//Managing and updating preferences in the user profile
date_of_data_collection = Recent_date_in_user_profile;
theSmallest_weight_userPrifile_should_have = 0.1;
number_of_days_since_last_visit = 20;
Daily_decrease_weight = 0.05;
Daily_increase_weight = 0.05;
TheLowestDuaration = 10;
for each Preference in User_Profile do
    //The following line will retrieve set of details about url such as date of visit, total time in all visit
    and user ID
    UserProfile = details about visited website;
    Update_wieght_for_url(Preference, UserProfile, date_of_data_collection,
    theSmallest_weight_userPrifile_should_have, Daily_decrease_weight, Daily_increase_weight,
    number_of_days_since_last_visit)
    currentWeight = Specific_user_urls.getTerm_frq();
    formatter = new SimpleDateFormat("yyyy-MM-dd");
    diffTime = 0; DifferenceIndate = 0; tempWeight = 0.0;
    for i <UserProfile.size() do
        log1 = UserProfile.get(i);
        dateInString = log1.getDayOfVisit();
        date1 = formatter.parse(dateInString);
        if (i + 1) $\geq$ UserProfile.size() then
            date2 = formatter.parse(date_of_data_collection);
            diffTime = date2.getTime() - date1.getTime();
            DifferenceIndate = diffTime / (1000 * 60 * 60 * 24);
            tempWeight = (currentWeight - (decreaseValue * DifferenceIndate));
            if (DifferenceIndate $\geq$ maxNumofDays) or (tempWeight $\leq$ smallestValue) then
                currentWeight = tempWeight;
                Delete(Preference);
            else
                currentWeight = tempWeight + increaseValue;
                Update(Preference);
            end
        else
            log2 = UserProfile.get(i + 1);
            date2 = formatter.parse(log2.getDayOfVisit());
            diffTime = date2.getTime() - date1.getTime();
            DifferenceIndate = diffTime / (1000 * 60 * 60 * 24);
            tempWeight = (currentWeight - (decreaseValue * DifferenceIndate));
            if (DifferenceIndate $\geq$ maxNumofDays) or (tempWeight $\leq$ smallestValue) then
                currentWeight = Specific_user_urls.getTerm_frq();
            else
                // This line to increase the weight.
                currentWeight = tempWeight + increaseValue;
            end
        end
    end
end
```

5.3.1.3 Forgetting Mechanism

Some users show no interest in some concepts and therefore, such concepts should be removed from the user's profile in order to provide more accurate results that do not contain old interests. There are some factors that indicate that a specific concept has become unwanted and needs to be deleted. As the work of Liu et al. [214] suggested, since the minimum time spent by a user on a webpage is between 10-20 seconds, and since this period reflects whether the user is interested in the webpage or not. Therefore, when a user spends more than the suggested time on a webpage, this means he/she is interested; otherwise, it means he/she is not interested. As a result, we established 10 seconds as our threshold and hence, when a user spends less time on a webpage than the threshold, it means that he/she is not interested in the webpage. Furthermore, for the URLs that have not been visited by the user, we suggested an initial forgetting mechanism that will use the current days since last visit in browsers (i.e. 20 days). Thus, if a URL has not been visited by the user for 20 days, then the interest will be deleted. The second factor, which is term frequency, involves decreasing the term value in time, in case the user does not revisit the URL, until it reaches our threshold (0.1); then it will be deleted by the deletion component. Algorithm 5 will be run after algorithm 4 just to make sure that all preferences' weights and dates fit with our determined thresholds.

Algorithm 5: Forgetting Concept from the User Profile

```
Data: User's Preferences
Result: Set of preferences without unwanted items
//This to delete any items with weight less than threshold or number of days since last visit greater than 20
for each Preference in User_Profile do
    if (Preference.getWeight $\leq$ 0.1 ) or (Preference.getLastVisit $\geq$ 20) then
        Delete(Preference);
    end
end
```

5.3.2 User Profile's Implementation for Semantic Similarity Experiment

This method is not as accurate as the previous ones; however, this way of collecting data has been considered for two reasons: i) to try to assess which provides better enhancement, sibling or semantic similarity relations, so the user profile is representing a secondary target in this experiment in contrast with 6.3 experiment; ii) there was a limitation of time for conducting this experiment and hence profiles could not be collected implicitly. In order to collect user preferences implicitly, we need to leave the tools (5.3.1.1 and *ManicTime*) in the user's machine for long period of time (e.g. a month) to be able to conclude user preferences. Thus, constructing ontological user profiles can be summarised as follow:

1. Participants register with our system.
2. Participants add 10 preferences in the first instance which reflect the most important topics that he prefers to read about.
3. Giving a score from 0.5 to 5, which reflects the topic importance to the user. These weights will be matched or converted to scale from 0.1 to 1.0. So, if the user gives a concept a 0.5, this means the score or weight for this concept will be stored as 0.1. But, when he gives a concept a 1.0, this means the score or weight for this concept will be 0.2, until he selects 5.0 which converts to 1.0 and reflects that this concept is very important to the user. However, 0.1 reflects that this concept is not important to the user.
4. Mapping user preferences to ODP concepts (branch of bioinformatics); each concept entered by the user will be combined with a concept from the ODP ontology which satisfied the highest cosine similarity score. This is done in order to exploit information acquired from ODP to enrich user profile with valuable information that can help to enhance the accuracy of the provided recommendations.

Algorithm (6) describes constructing an ontological user profile in more detail.

Algorithm 6: Managing User Profile for Semantic Similarity Method

```
Data: Concepts entered by user and their weights
Result: List of preferences
//Adding Preferences to the user profile
for counter <number_of_concepts_entered_by_user do
    add_preference_and_their_weights(); //This step to add user preference, and then matched
    with most similar concept of ODP to exploit semantic similarity relation
    convert_term_or_concept_weight_to_defined_criteria(); //This step to convert user given
    weight to scale from 0.1 to 1.0
    counter++;
end
//The following loop will go through all user's preferences and map them to ODP concepts
for each Concept in user_List_of_concepts do
    V2 = Convert_user's_ Concept_to_vector(); //Convert concept into vector to calculate similarity.
    The following loop goes through user's concepts.
    for each Ontology_Concept in ODP ontology do
        V1 = Convert_ontology_concept_to_vector(Ontology_Concept); //Convert ontology concept and
        its description into vector to calculate similarity.
        sim = V1.getCosineSimilarityWith(V2); //Calculate cosine similarity between two vectors.
        //This following condition is to find out the best matched concept from the ODP ontology and
        the threshold (0.15) has been selected based on several runs, where this threshold represent the
        best match threshold.
        if sim >maxSim and sim >0.15 then
            maxSim = sim;
            BestMatch = Ontology_Concept;
        end
    end
    Update_preference_Record_in_DB(BestMatch, Concept, maxSim)); //This step will update user
    record in the database and match it with the best match element or concept from ODP ontology
end
Call (algorithm 7) //This step to map user's preferences with inferred semantic network
```

5.3.3 Method for Mapping Semantic Concepts to User Profiles

The method for mapping the most similar concepts from the semantic network to the user profiles involves a number of stages. First, each concept from the user profiles goes through direct mapping with semantic network concepts. In this step we tried to filter all concepts that already existed in the inferred semantic network to avoid any further cosine similarity calculation to find the most similar concept from the semantic network. If the same concept in the user profile was found in the semantic network, then this concept is matched with the concept found in the semantic network, and it is enriched with extra information and relations (e.g. sibling relation or semantic similarity) that are gained by the semantic network as a result of information overlapping among different resources. Otherwise, the cosine similarity will be computed between each concept of the user profile with semantic network concepts. This will be performed by taking the content of the surfed website and the definition of the matched concept from the ODP ontology, which we have used to create our ontological user profile, for each user's preference. Then, adding them together and preparing them to calculate the cosine similarity between them and the name, label and comment of each concept in the inferred semantic network. Then each user profile concept will be mapped and synchronised with the concept in the semantic network that satisfies the highest cosine similarity score with the user's profile. This time, the cosine similarity will be computed between the user profile concept's description and the semantic network concept's description.

For instance, let's assume that a specialist is interested in a gene called GO_0000259, the "intracellular nucleoside transmembrane transporter activity"[1], so when we perform a direct mapping between this gene and our inferred semantic network concept, we will find that our semantic network contains the same concept. In this case there is no need to do a further calculation or perform cosine similarity, and we will make a direct match between the two concepts and exploit the extra information that can be gained from the semantic network concept. If we exploit the sibling relation, we will consider the concept PR_000000549, which is a protein called "growth/differentiation factor 7 isoform 1 cleaved form"[1] as its sibling, which is the first sibling of the concept GO_0000259. However, if we try to exploit the semantic similarity relation, then we will consider concept GO_0000100, a gene named "S-methylmethionine transmembrane transporter activity"[2], which satisfies the highest semantic similarity score with the main concept GO_0000259.

As another example of the second case, when the specialist is interested in a concept that does not exist in our inferred semantic network, then we need to consider a cosine similarity score instead of direct matching. For instance, if someone is interested in gene GO_0000101, "sulfur amino acid transport"[2], this

[1] http://pir.georgetown.edu/pro/

[2] http://geneontology.org/

gene is not included in our inferred semantic network. So, we calculate the cosine similarity between this gene and all concepts in the inferred semantic network. Thus, we find that the most similar concept to the GO_0000101 is a gene called GO_0000096, which is called “sulfur amino acid metabolic process”[2], so this gene will definitely be considered in the recommended articles. If we exploit the sibling relation, then the first sibling of GO_0000096, which is called GO_0000103, “sulfate assimilation”[2], will be considered by the recommended articles that have sibling relations to the concept that satisfies the highest cosine similarity score with GO_0000101. But if we decide to exploit the semantic similarity relation to provide recommendations, then the gene called GO_0000097, which is named “sulfur amino acid biosynthetic process”[2], will be considered to recommend articles that are semantically similar to the concept which satisfies the highest cosine similarity score with GO_0000101. Algorithm 7 discusses in more detail the steps that were considered to map user preferences to the semantic network and create an adaptive sub-network to each user based on his/her preferences.

Algorithm 7: Mapping Semantic Network with User profiles

```
Data: User's preference and semantic network's concept
Result: A user profile mapped with the semantic network (sub-network for each user)
userInterests = getAllUserInterestedTopics(); // This is to retrieve all interests in the user profile
SemanticClasses = getAllSemanticNetworkClasses(); // This is to retrieve all semantic
  network's concepts
Double HighestMatch = 0.0; // This will save the highest match score for each concept in the user
  profile
// This loop will go through all user's interests
for each Interest in userInterests do
    // This loop will go through all classes in the semantic network
    for each class in SemanticClasses do
        // The following line is to check if user preference already exists in the inferred semantic
          network
        if (Interest = class ) then
            map(Interest, class); // This is a direct mapping between user interest's and semantic
              network's concept
        else
            Simi = ComputeCosineSimilarity(Interest.Content, class.Content); // This is to
              compute cosine similarity between user interest's and semantic network concepts
            if (HighestMatch <Simi) then
                HighestMatch = Simi; // This is to assign the cosine similarity socre as the highest
                  match value between user interest's and semantic network concept
                map(Interest, class); // This is to map between user interest's and semantic
                  network's concept that have highest cosine similarity score
            end
        end
    end
end
```

5.4 Reasoning with and Extracting Information from Multiple Resources

This stage will be concerned with the core focus of our research. By performing this step, we will be able to extract, reason and exploit a variety of information gained as a result of information overlapping between different processed resources. In this section, we will discuss extracting, reasoning through multiple resources and terminating by sending the discovered information to the semantic network algorithm in order to represent the inferred relations and information. This information is to be mapped to the user profile and then exploited by the recommender system in order to provide a personalised recommendation to each user based on his/her preferences stored in the profile.

The extraction process represents the first step of discovering semantic relations. Since, we determined the information that should be extracted from each file or resource based on studying and analysing the structure of these resources. Then, we represented these resources in a single format, which is OWL, to deal with their data easily and efficiently. As a result, we extracted classes such as subClassOf, equivalentClasses and someValuesFrom. This extraction from different resources led us to discover some rich relations gained as a result of information overlapping, where some of these relations can be classified as semantic relations. The extraction process will go through several steps, cooperating with the reasoner to extract semantic relations and hidden associations that can be found between different bioinformatics resources. Dealing with these resources with different structures is not an easy task because of the inconsistencies between their structures and the type of relations that exist in each resource. To overcome these difficulties, we have designed our SPARQL method to cooperate with the Jena reasoner by using the Jena framework, which is one of the most well-known frameworks that can handle files with different formats such as OWL and RDF [212].

In our extracting, reasoning and designing semantic network, we have not applied these methods in all datasets due to the limitations of computer capabilities. We tried to run these processes on several machines, but no machines were able to handle the discovered semantic relations. We decided to divide our dataset into several sub-files and perform the extracting, reasoning, inference and designing processes. Our dataset before we divided it into several parts was 94.5MB GO, 48.7 MB PO, 41.4 MB BLD, 1.55 MB Wikipedia terms, 126 KB ODP ontology. Based on several runs performed on this dataset, the maximum sizes our machines were able to handle are three sub-files of GO with 1.38 MB, three sub-files of PO with 1.38 MB, two files of BLD ontology, which represent a DNA directory and an education directory with 349 KB for each; and Wikipedia and ODP of the same sizes. We tried to run these processes on the following machines: Windows 7 64-bit with a 512 SSD drive; 8GB RAM and Core i7 with 2.80 GHz processor laptop; Windows 7 64-bit with 500 HD drive; and 8 GB RAM Core i3 with 3.10 GHz processor PC. Finally, we ran these processes on a Linux server with 120 HD, 33 GB RAs Xeon CPU ES-2470 with 2.30 GHz processor, which is able to handle our dataset by extracting, reasoning, and inferring semantic relations and hidden associations, then design the semantic network successfully.

The process of extracting from multiple resources will be explained in detail in the following steps:

1. The extraction algorithms were designed to create an inference model in the Jena framework for each OWL file, which contains data selected from each resource. The selected data excluded specific types of data, which were determined by the developer in order to avoid selecting extra data that would be time-consuming to process and reason. This was also done to overcome all of the inconsistencies between resources that could occur as a result of retrieving all of the data from the processed resources, which contained different types of relations that might be incompatible with each other.

2. After the data are extracted from all of the resources, the models will be assembled in a single model. Accumulating different inference models in a single model will contribute to providing an initial solution for the problem of resources inconsistencies. To this end, the inference model will be ready to be sent to the reasoner to discover new relations and information that may not exist in the original resources.

3. The inference model will be sent to the Jena reasoner, which allows us to

perform reasoning through our extracted data or to discover new relations or hidden associations between the extracted data. It is also equipped with specially created Jena rules (i.e. Jena rules are classified into two types: forward and backward rules [206]. The Jena rules that are applied in our approach were discussed in detail in section 4.2.3). These rules were designed to infer specific types of relations, such as *siblings, grandSubClassOf etc.* Moreover, this reasoner accepts built-in custom rules, such as our semantic similarity method. It consists of a Jena rule which is calling an algorithm that compares the descriptions of the two classes of our inferred data. So, in the case that the similarity score between them exceeds the threshold (the threshold was decided based on multiple runs to find the most similar cases between the different classes), then the two classes will be classified as semantically similar classes.

4. Finally, an array of inferred models will be sent to the semantic network designing algorithm 9. This algorithm will formulate these relations into a semantic network to be exploited by different user profiles based on direct matching and cosine similarity between user interests and semantic network concepts. Algorithm 8 describes the extraction and reasoning process.

Algorithm 8: Extracting and Reasoning Algorithm

```
Data: Selected Data (e.g. class, subClassOf, comment,etc.)
Result: List of inference models
Prepare_Jena_Reasoner(); // This to reason through all models after read them.
Declare ArrayList of models of type InfoModel // This will contain all information selected by
SPARQL query
Declare Inference repository of type Object // This will keep all arrays to be exploited when
semantic network created
for each OWL_file in OWL_List do
    Create SPARQL Query(term);
    SELECT ?class ?subClassOf ?comment ?label ?type ?someValueFrom ?objectProperty
    ?equivalentClass //This shows all contents that have been selected from each OWL file
    WHERE
    ?class rdf:type owl:Class . //This will extract only classes that of type owl.
    OPTIONAL{ ?class rdfs:subClassOf ?subClassOf .}. // This line is to find all subclassOf in
    case of class has subclassOf, where this is optional
    OPTIONAL{ ?class rdfs:label ?label .} // This line is to find all label in case of class has
    label, where this is optional
    OPTIONAL{?class rdfs:comment ?comment .} // This line is to find all comment in case of
    class has comment, where this is optional
    OPTIONAL{?class rdf:type ?type .} // This line is to find all types in case of class has type,
    where this is optional
    OPTIONAL {?class owl:equivalentClass [ a owl:Class ; owl:intersectionOf [
    rdf:rest*/rdf:first ?equivalentClass ]] .} // This line to retrieve all classes in case if class
    consists of complemented classes.
    models.add(selected data) // This line to fill model list with selected information from each file
    read.
end
Jena Reasoner (models,rule file); // This line will send semantic rule file and set of models to the
reasoner to infer new semantic relations and hidden association between different models
Repository.StoreAll_InferedData(); // This to store all inferred triple in order to exploited by
semantic network.
```

5.5 Creating the Semantic Network

To represent our semantic network as described in section 4.3 with all inferred relations and information, the Dijkstra [215] algorithm was modified to fit with our resources. This algorithm was designed to calculate the shortest path between two points in a graph. This method is well known and has been used for several approaches because of its accuracy and speed in calculating the shortest path between any two classes, points or concepts in a graph.

We altered this method to calculate the shortest path from the class *"Thing"* instead of calculating the path from any point in the graph. There were two reasons for this change to Dijkstra's algorithm: i) to assign weight to the edges (i.e. links) that connect different concepts in our semantic network, where the links' weight started at 1 and the number decreased by 0.25[1] whenever the concept stayed one

[1] We have considered this number as suggested in [216], the reason behind selecting this particular value is that the average of neighbours for each node in our semantic network equals 4, this means one divided by four.

step away from the class *Thing*. This calculation will help us to know how far the concept is from the class *Thing*. As a result of the first reason, ii) we can determine the distance between any pair of concepts in our semantic network. This can be done by calculating the total number of edges' weights from each concept to the class *Thing*, which can be done by using equation 4.4. Performing these steps allows us to calculate the semantic similarity between any two concepts by using equation 4.5. Finally, the semantic network will be stored in an XML file and exploited by the user's profile. This is done after it overcomes three levels of challenges stemming from inconsistencies that occur as a result of different structures and relations included in each resource. The following steps describe the stages, followed by our semantic method to overcome challenges stemming from inconsistencies that may occur during the reasoning process and representing the inferred semantic network.

- Our semantic method will filter the selected data from different resources by applying a SPARQL query that selects specific relations and information to avoid considering any unwanted relation or information that may complicate the inference process.
- Then, our semantic-based method will remove any duplicate classes and keep only unique classes.
- Finally, our method will check whether the inferred data already exists in the class or not.

Algorithm 9 describes all steps considered to design our inferred semantic network in more detail.

Algorithm 9: Designing Semantic Network

```
Input: Triples' Data
Output: Semantic Network
List_of_SPARQL_Query_Results(); //This represents retrieved classes form SPARQL query, where SPARQL query used to overcome the first level of inconsistencies between different resources and to avoid extracting unwanted information or relations which may disturb representing the inferred semantic network process.
Semantic_Network_Designer(); //This class is to design semantic network
ArrayList<Semantic_Network_DS>classes;//Define arraylist of classes of type Semantic_Network_DS
//This loop will go through all classes returned by SPARQL query

for each class in List_of_SPARQL_Query_Results do getClassDetails(class);//This will take each class in order to formulate its entities
DetermineRoot();//This will make class "Thing" as subClassOf for all orphan classes
ConstructClassDS();//This will create a unique class with its' contents such as class's name, subClassOfList,etc.
// The following condition is designed to overcome the second level of inconsistencies between resources, which done by removing duplicated classes and keep unique classes with their properties.
    if !classes.contain(class) then
classes.add(class); //This will add class to list of classes after all preparations
    end if
end forReturn classes; // This represents the pure semantic network
// This loop will go through returned classes
for each class in classes do // This loop will go through triples which stored in the repository (i.e. result of algorithm 1)
    for each triple in Repository do
        if class = triple.getSubject() then
// The following condition is designed to overcome the third level of inconsistencies between classes where it double check whether the inferred data are already exists in the class or not.
            if !class.contains(enrichWithInferedRelations()) then
class.enrichWithInferedRelations(); // This is to match class with inferred data
            end if
        end if
    end for
end forReturn enrichedClasses // This will return all classes after enrichment
// This will find out subclasses and suerclasses for all classes and prepare them, then send them to Dijkstra algorithm
List SemanticClasses = DetermineAll_Parents_&_Children(enrichedClasses);
Map SemanticPaths = Dijkstra(SemanticClasses); // This is to call Dijkstra to calculate the shortest path between the start node, which "Thing" and any other node in the semantic network
Semantic_Network_DS Start = DetermineStartPoint(); // Start point is usually assigned with Class "Thing", and it of type Semantic_Network_DS
 CalculateVertics(Start,1); // This receives the start point and its level which usually assigned by 1 as a start level
ComputeAllEdages(); // Compute the edge between two concepts
ComputePaths(); // This function has been adjusted to usually start by "Thing" to be the start vertex
Return (Concept, distanceToStartPoint); // This will return all concepts in the semantic network and their shortest distance to the start point "Thing"
CalculateDistance(concept1,concept2); // This is an optional choice, where it can be run at any time, if we need to calculate distance between two concepts, then it should apply equation 4.4
SemanticSimilarity(concept1,concept2); // This is also an optional choice, where it can be run at any time, if we need to calculate semantic similarity between two concepts, then it should apply equation 4.5
// The following loop will store each class of the semantic network as element in the xml file, where it will avoid any duplication to the classes and it represents the final level of handling inconsistencies between classes.

for each class in Semantic_Network do
The following condition, is to check whether this classes added to the xml element to be represented or not

    if !xmlElements.contain(class) then xmlElements.add(class); // This is to add the new class to xml elements
//The following condition is to add only extra details (such as semantic distance) in case of class has been added as xml element

    end if
    if xmlElements.getElement_Details()!= class.Details() then xmlElements.getElement.add(class.Details());// This is to add extra details about to the xml element.
    end if
end for
StoreAll_Semantic_Network_details_in_XML(xmlElements); // This will represent all semantic network's details in an XML file to be exploited by other components in our approach
```

5.6 The Semantic Similarity Method Implementation

The semantic similarity inference method procedure[1] involves three stages: (i) the Jena reasoner will call a semantic custom built-in rule to run through every inferred concept. This rule will then call for a similarity method, which will filter similar concepts by comparing the description of each concept with all inferred concepts. It then calculates the similarity of their descriptions and considers any concept satisfying a threshold (for our underlying ontologies, this was experimentally determined to be 0.6) as an initially similar concept. In step (ii), it calls for another function to calculate the semantic similarity between the compared concepts. Step (iii) involves crossing the two scores (i.e. concept similarity and semantic similarity scores), and if they register above a specified threshold (this was experimentally determined to be 0.3), they will be considered semantically similar concepts. Tables 4.1 and 4.8 provide the full details of the rules designed. In practical terms, and for our experiment the thresholds were identified based on several runs, and the scores that satisfied the most similar concepts were picked to represent our semantic similarity thresholds for the similarity of various concepts from different bioinformatics resources. As these resources are not typically used together, experimentation was the only way to determine the more suitable thresholds. The semantic similarity method can be distinguished by its ability to calculate semantic similarity with a reasoning process, and this may help to infer new semantically similar concepts that may not appear in the typical way. An additional distinguishing feature is its ability to handle multiple resources with different structures and extract relations and information and subsequently employ them to enhance the accuracy of the recommendations. Algorithm 10 describes this method in more detail.

[1] Our semantic similarity method did not concern the time taken to calculate the semantic similarity between concepts. It was concerned with the quality and accuracy of the semantic similarity results that were discovered between concepts. Thus, it may appear that our semantic similarity method is slower than other semantic similarity methods. However, it is more accurate, and the level of accuracy can be measured based on the assessment of the subjective participants when they test our prototype system supported by our developed semantic similarity method.

Algorithm 10: Semantic Similarity Method

```
Data: Compared Classes and their comments
Result: Concepts have semantic similarity
BuiltinRegistry.theRegistry.register(new BaseBuiltin()); //This to register method's name in the reasoner registry
getName(); // This line to return the name of custom built in rule.
bodyCall(Node [] args,length,Rule_Context) // This to define all parts in the semantic rule
checkArgs(length, context) // This line is to check receive arguments and their length
final Node n3 = getArg(0, args, context); // This contains the first class
final Node n4 = getArg(1, args, context); // This contains the second class
final Node n1 = getArg(2, args, context); // This contains first class's comment
final Node n2 = getArg(3, args, context); // This contains second class's comment
// This to check all comments are string
if n1.isLiteral() && n2.isLiteral() then
    FirstComment =Split(n1); // This will split first comment into arrays
    SecondComment =Split(n2); // This will split second comment into arrays
    sort(FirstComment,SecondComment); // This will sort the two arrays
    int count_firstComment_length = 0, count_SecondComment_length = 0 // These to count number of words in the arrays
    // This condition to keep loop continue through all concepts
    while (count_firstComment_length <FirstComment.length && count_firstComment_length <SecondComment.length) do
        // Compare the two Comments
    end
    // This following line is to calculate similarity score between two concepts
    Similarity_Score = (2.0f * counter_Same_Words) / (FirstComment.length + SecondComment.length);
    // This condition to filter similarity scores.
    if (Similarity_Score >0.6) then
        twoWords = (n3,n4); // This type has been created to save two words together
        Similarity_OF_twoComments = map (twoWords, Similarity_Score) // Save similarity concept and similarity score
    end
end
Return True;
List SemanticClasses = Call_Constructed_Semantic_Network(); //This will return list of class and their scores of shortest path to "Thing", after it reasoned through different resources and represented the inferred data on the fly
SemanticClasses = Calculate_Semantic_distinces(Similarity_OF_twoComments, List SemanticClasses); // This function will receive list classes and list of classes with their distance score from class "Thing"
// This loop will go through concepts and their similarities
for (each twoWord in Similarity_OF_twoComments) do
    String word1 = twoWord.w1; // This retrieve the first class or concept
    String word2 = twoWord.w2; // This retrieve the second class or concept
    double word1Value = List SemanticClasses.get(word1); // This retrieve the semantic distance value of the first class or concept
    double word2Value = List SemanticClasses.get(word2); // This retrieve the semantic distance value of the second class or concept
    double simanticSimilarity = 1 / ((word1Value + word2Value ) + 1); //calculate the initial semantic similarity score
    double conceptSimilarity = newHash.get(twoWord);
    // The following line will calculate the final semantic similarity score
    double newSemantic_Similarity_Value = simanticSimilarity * conceptSimilarity;
    //This condition is to filter most relevant classes
    if (newValue >0.3) then
        newHashNew.put(twoWord, newSemantic_Similarity_Value);
    end
end
StoreAll_Semantic_Network_details_in_XML(); // This will represent all semantic network in XML file to be exploited by other component in our approach.
```

5.7 Component for Automatic Updating of Semantic Concepts

As mentioned previously, the importance of this method is that it adapts to changes and updates that may happen to the online resources. These updates can add extra information to the concepts of our processed resources that contributes in enhancing the quality of the provided recommendations that support bioinformaticians to find what they are looking for without consuming their time and efforts. Moreover, it represents a support method to our method which designed to represent the inferred semantic relations and associations in a semantic network. So, this method will ensure that most of the data in the semantic network are up-to-date.

The update component uses several steps to perform its task, which are as follows: i) the update component is equipped with a trigger that fires every month, contacts the PO and GO websites and then downloads and unzips their ontology files. ii) Then it extracts the same data that have been previously extracted to formulate our semantic network, such as *class, subClassOf, comment and label.* iii) It performs a reasoning process on the extracted information to compare it with the old concepts and check whether these concepts have updates or changes. iv) If they have updates or changes, then the component will replace the updated concepts and their relations with the old concepts that exist in the network, v) after which an up-to-date semantic network will be created. Algorithm 11 describes the aforementioned steps in more detail.

Algorithm 11: Automatic Update Component

```
Data: Ontologies from PO and GO websites
Result: Updated concepts in the Semantic Network
// This is to check whether last update pass a month or not
if getTimeTheCurrentTime() $\geq$ GetLastUpdateDate() + 30 then
    RunUpdateTriger(); // This represents start of running update component
    Download(GeneOntology, ProteinOntology); // This is to download gene and protein
    ontologies to compare them with the exists version
    Result = run(Extracting and Reasoning (GO, PO)); // This is to run extraction and
    reasoning method which discussed before in algorithm 8
    subnet = Design(Semantic Network (Result)); // This is to apply the first 26 lines of
    algorithm 9 to design a sub network
    subXML = Represent_in_XML(subnet); // This will represent the designed sub semantic
    network in XML file in order to compare it with the old version of GO and PO that were included
    in our semantic network
    // This loop will go through all classes in sub-semantic network
    for each class in subXML do
        // This loop will go through all classes in the semantic network
        for each oldClass in SemanticNetwork_XML do
            // This condition is to check whether any class's content has changed in gene or protein
            ontologies, since last update performed on the semantic network
            if (class = oldClass) and (class.Content $\neq$ oldClass.Content) then
                replace(oldClass with class); // This is to replace the old class with new one
            else
            end
        end
    end
    UpdateTime = getTheCurrentTime(); // This is to save the update time
end
```

5.8 Content-based Recommendation Service

The enhancement of recommendation services is reflected the success of our developed semantic-based methods. This enhancement is done by exploiting the data and relations (i.e. sibling and semantic similarity) collected by the inference method that is applied for different resources. The inference method is able to extract semantic relations and hidden associations between these resources and then represent the gained relations and information as a semantic network. Then, it will connect semantic network concepts with the most similar content in each user profile in order to enhance the accuracy of the provided recommendations. To achieve this target, a prototype system was developed that helps bioinformaticians find articles in the BMC corpus. The Lucene Search Engine[1] was used for indexing and retrieving articles from the BMC corpus. A set of ontologies (i.e. GO, PO, ODP and BLD) and Wikipedia were employed to extract semantic information for users' query enrichment. A method was developed that is able to collect user preferences automatically and implicitly, and then it calculates similarity between the ODP ontology and the user's preferences to construct an ontological user profile. Figure 5.1 shows different component of our prototype recommender approach as well as the different resources that will be processed to achieve the main goal of this research . A set of re-ranked articles will be returned and organised based on their similarity to both the user's preferences and semantic enrichment. Moreover, in the case that an article of the top 30 re-ranked articles exists in the recommendation list, then its priority score will be updated (increased), and it will be shown in the recommendations list of the user based on his/her preferences; otherwise, this article will be added to the list of recommended articles and will be shown in the recommendation list. This system gives users the opportunity to narrow down recommendations by selecting a specific interest to receive recommendations exclusive to the elected preference. Algorithm 12 provides an outline of the high level process performed by the prototype system.

[1] http://lucene.apache.org/core/

Algorithm 12: Content-Based Recommendation Services

```
Data: User's Query and User's ID
Result: List of Recommended Items
Submit(query);//To receive query.
Enriched_query = Enriched_with_required_relation(query,User_ID); This to enrich user's query with
semantic relation concept.
lucene_Search_engine(Enriched_query,100); //This will send user's enriched query and number of hits
results.
//The following loop will go through all articles.
for returned_Article do
    //The following loop is to read files.
    while counter <returned_Article.length do
        Array_of_Strings = read(returned_Article);//Read file and store it in array.
    end
    V2 = Convert_query_result_to_vector(Array_of_Strings); //Convert file into vector to calculate
    similarity.
    Get_User_preferred_Concept(User_ID); //Retrieve all user's concepts.
    //The following loop goes through user's concepts.
    for each User's Concept do
        SemanticEnrichedConcept= (userConcept.Description + SN_enrichment);
        V1 = Convert_User_concept_to_vector(SemanticEnrichedConcept); //Convert user's concepts
        description and semantic enrichment into vector to calculate similarity.
        sim = V1.getCosineSimilarityWith(V2); //Calculate cosine similarity between two vectors.
        Document_Simi_Score +=termWeight() * sim; //Total similarity for each file.
    end
    Document_final_Score = Document_Simi_Score * lucene_Score;
    hitsMap.put(queryFilePaths,Document_final_Score); //Fill Hashmap with file paths and score of
    similarity.
end
//The following loop will add and update and show recommendations on top 30 preferred articles.
for each Result in hashMap do
    Index = 29; //The following loop will add and update preferred articles.
    add_to_userprofile(Result);//Save recommendaed articles in the user profile.
    Index –;
    if Result in 30th preferred Articles then
        Update_Userprofile_score(Result);//Update document score.
        Show(Recommended_Articles); //This shows recommended articles that already exists before,
        based on similarity with user preferences and similarity with exploited semantic relation
        (Sibling or semantic similarity).
    else
        add_to_userprofile(Result);//Save recommendaed articles in the user profile.
        Show(Recommended_Articles); //This shows recommended articles that already exists before,
        based on similarity with user preferences and similarity with exploited semantic relation
        (Sibling or semantic similarity).
    end
end
```

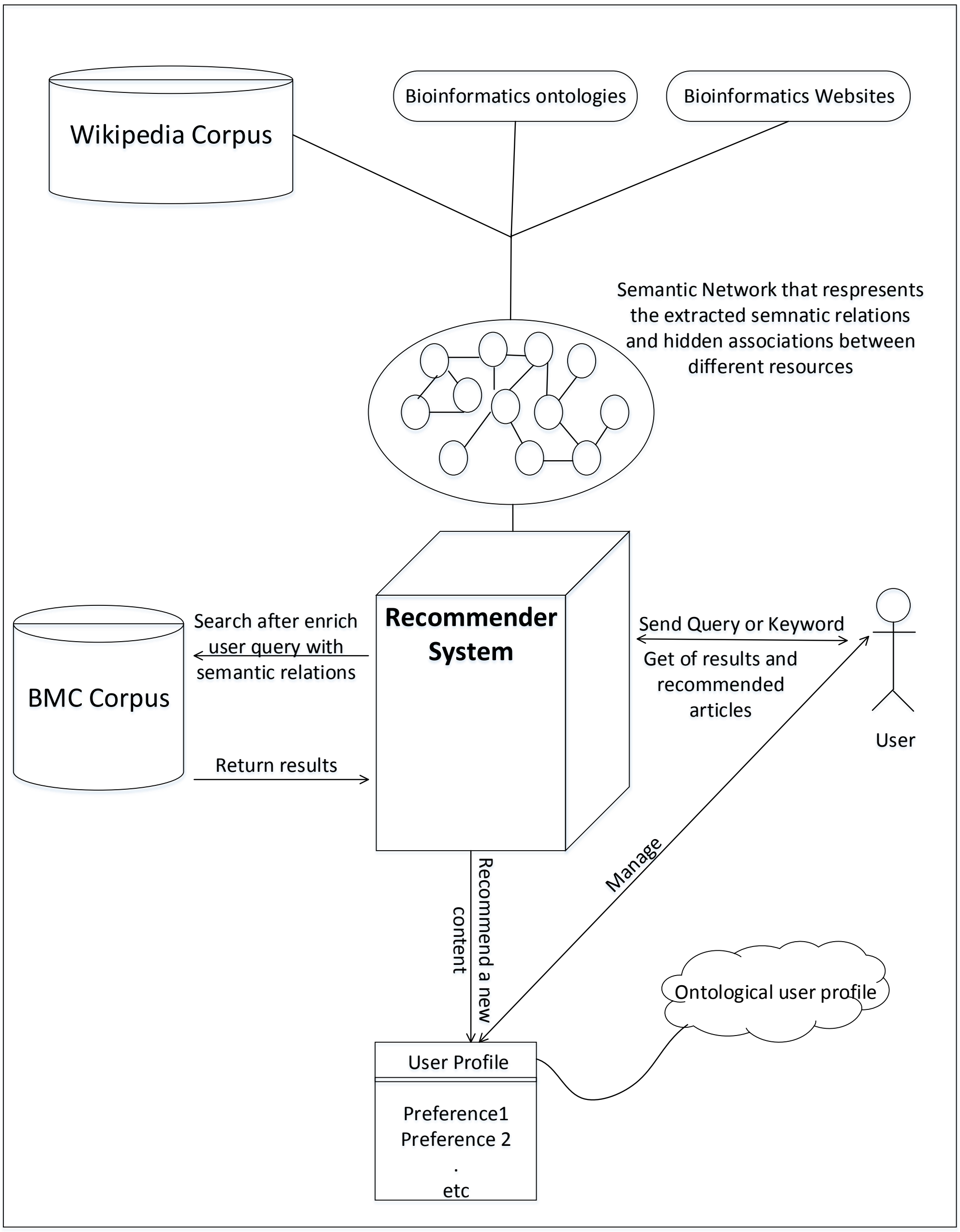

Wikipedia Corpus
Bioinformatics ontologies
Bioinformatics Websites
Semantic Network that respresents the extracted semnatic relations and hidden associations between different resources
BMC Corpus
Search after enrich user query with semantic relations
Return results
Recommender System
Send Query or Keyword
Get of results and recommended articles
User
Manage
Recommend a new content
Ontological user profile
User Profile
Preference1
Preference 2
.
etc

Figure 5.1: Project General Structure.

5.9 Conclusions

To conclude, this chapter provided our prototype system that has been used to apply our suggested hypotheses in order to achieve the goal of the thesis, which is to support specialists in a specific domain, such as bioinformatics, by providing them with recommendations based on their preferences, search activities and discovered relations between concepts in semantic-based resources (sibling or semantic similarity). Moreover, it discussed the different components, methods and tools that were used/developed to implement our main semantic-based method as well as our recommender approach. Furthermore, it demonstrated the implementation steps of each method and how these different methods work together to enhance the accuracy of the recommendations provided.

The next chapter will discuss the evaluation of the methods that we developed using a set of experiments that were designed to measure the level of success achieved by our approach compared with other comparative approaches. In addition, it will discuss the results and reasons behind the strengths or weaknesses of our recommender approach.

Chapter 6

Evaluation of Semantic-based Methods

6.1 Introduction

This chapter discusses the evaluation undertaken to assess the success of running our prototype system over our prepared dataset. We measured which of our relations could support our recommender system to provide better recommendations compared with the other comparative approaches in this assessment. Thus, we conducted two experiments on our dataset. The first experiment assessed how much our contribution enhanced the recommender system when we exploited the sibling relations to support specialists in specific domains, such as bioinformatics, with accurate recommendations. The second measured the utilisation of our recommender system when we applied our contribution to exploit semantic similarity relations to enhance the precision of the provided recommendations.

Moreover, we have evaluated different aspects of our system against other systems. This evaluation allows us to compare the accuracy of our recommender system compared with systems discussed in the literature and baseline. Also, it helps to compare different functionalities included in our recommender system against each other such as the performance of our recommender system with/without user profile or semantic relation exploitation. Moreover, it allows us to test which of our exploited semantic relations (sibling and semantic similarity) can support our recommender system with more accurate recommendations. This could help us to have an idea about the extent of enhancement of recommendations that could be achieved as a result of applying our method in our recommender approach.

As we discussed earlier in section 3.4, recommender systems can be evaluated by three types of evaluation methods (offline, user-centric and online). Thus, we have undertaken a user-centric method to evaluate our bioinformatics recommender approach, by searching for participants who specialise in the field of bioinformatics. This will allow us to draw an accurate conclusion about most weakness and strength points in our recommender system.

6.2 Experiments Evaluation Metrics

In both experiments, 6.3 and 6.4, we applied a set of evaluation metrics to measure the level of enhancement that could be achieved when we exploit our discovered semantic relations (i.e. sibling and semantic similarity) in our prototype system and compare them with the other approaches. Thus, we applied **Precision@N**, which was taken over top@ 5, 10, 15, 20, 25 and 30, since it is difficult for users to rate all the provided results; and **Mean Average Precision** [202] to assess our method in terms of classification accuracy in comparison with the other comparative approaches in both experiments. So, **Mean Average Precision** [202] was also applied to ensure that all results gained by **Precision** at **N** are accurate, where these metrics are useful for recommender systems [203], especially for recommender systems with a pre-ordained nature [204]. Moreover, we applied **Mean Absolute Error** to assess the performance of our approach in terms of the predictive accuracy on the provided recommendations when considering any of the discovered relations. Therefore, the former metrics' methods and their results will be discussed in more detail in the following sections.

6.3 Experiment 1: User-Centric Evaluation for Bioinformatics Recommender Services

This experiment was conducted with a set of participants who specialised in bioinformatics or biological science to assess the performance of our recommender approach in enhancing the preciseness of the provided recommendations on articles in the BMC corpus. This was done by applying the developed semantic techniques over multiple bioinformatics resources (ontologies and corpora) and employing the inferred relations, such as sibling relation, to enhance the precision of the provided recommendations.

The evaluation of our method was conducted by using 30 human participants who were experts in bioinformatics[1]. Each participant was assigned to groups that interacted with an evaluated approach (the comparative approaches in this experiment will be described in detail in section 6.3.4). He/she was asked to interact with his/her assigned system by performing five tasks (appendix E.2). This provided us with accurate results about the five approaches, which helped to assess our approach in contrast with the other approaches. The data were collected and stored based on four stages:

(i) Meeting with each participant and installing the plug-in (i.e. 5.3.1.1) and *ManicTime* in his/her machine.

(ii) Collecting data from the participant's machine and creating the ontological user profile.

(iii) Collecting data that represent user interactions (clicks, rates, etc.) with the systems while performing the five assigned tasks.

(iv) Collecting the questionnaires (appendix E.1), which reflect the level of satisfaction that participants have with the provided recommendations and with regards to the used recommender system.

[1] We have selected only this small number of subjects because conducting user-centred evaluation in the field of recommendation is known to be difficult and expensive [189].

6.3.1 Experiment Goal

There are three hypotheses that represent the main goal of this experiment:

- H1: Considering multiple resources and extracting semantic relations (e.g. siblings) and associations between them to enrich the dynamic ontological user profile and user query can enhance the recommendation and improve the retrieved result.
- H2: The use of the automatic adaptive profile that manages user interest changes, provides more accurate recommendations.
- H3: Narrowing the topics in the user profile into a single topic will contribute to more accurate recommendations.

Thus, this experiment is to assess our recommender approach in comparison with other recommender approaches when applying the three aforementioned hypotheses. Also, we will determine the level of preciseness enhancement for the recommendations with and without considering the semantic enrichment from multiple resources and user profile tailored from different resources to be applied in our approach. Moreover, we will compare our approach with a general approach such as Google and with another approach taken from the literature [159]. We also intend to measure the re-ranked results based on user profiles as to whether these results are acceptable to the user.

6.3.2 Experiment Participants

This experiment was conducted with students from the School of Biological Science at the University of Essex. These students have specialised in different branches of the biological studies such as medical microbiology, molecular medicine, biotechnology, medical biochemistry, biological science, virology and bioinformatics. Even though this experiment should concentrate on bioinformatics researchers only, however, due to the difficulty of finding participants who specialised in bioinformatics, had to find participants from relevant field who have experience in bioinformatics. The participants for the study worked with terminology from the domain of bioinformatics. Thus, they have the experience to assess whether our approach enhanced the preciseness of the provided recommendations for the retrieved articles. The study included 30 participants, and they were from both sexes (male and female).

6.3.3 Experiment's Steps

The experiment run through several steps described as follows:

(i) The plug-in (5.3.1.1) and *ManicTime* were installed on participants' machines. Each surfed content, bookmark and click will be stored in the database with the user number to distinguish between participants. These tools were installed on the users' machines after explaining that all information will be used for research purposes and will not be exploited for any other purpose. Moreover, all tools installed are not able to extract encrypted information due to security restrictions that are applied to the secure website such as e-mail and bank websites.

(ii) A method was designed to filter the surfed contents collected from users' machines during the 30 days, which is the duration that we left our tools observing users' interactions. This method filtered the data day by day and dealt with the data collected daily from the users to calculate the weight and priority for each item in the user profile. Our approach updated the users' profiles by increasing or decreasing the priority of different concepts based on their weight or adding a new concept if the users became interested in a new one, and all such processes were done automatically. Section 5.3 provided more details about constructing and managing the ontological user profile.

(iii) After collecting data, participants were divided into five groups and asked to perform five tasks (appendix E.2) with our approach and the other compared approaches.

(iv) The recommendations process in this experiment followed this procedure:

  - The specialist read the assigned task (appendix E.2). Then he/she submitted a query to the system, which enriched with the sibling concept to the most similar concept stored in the user profile that matched his/her query (except when the system did not have a user profile where the query was enriched with the sibling concept to the most similar concept from the inferred semantic network).

- The results retrieved from the Lucene search engine was organised by similarity between specialist's query and his/her stored preferences.
- In the same page of the retrieved results, there was a link called "See Recommendations". So, whenever the specialist clicked on this link, a list of top 30 recommended articles appeared, containing articles related to the concept that had a sibling relation to his preferences and organised based on their relevancy to his query.

Then, all the assigned tasks in appendix E.2 were repeated. The reason for repeating these tasks was to ask the specialist to select based on his/her preferences to get recommendations on it (the participants who were in the system based on only semantic enrichment also chose based on their preferences, but their selection was not applied to demonstrate the support of semantic-based method when using the user profile). This time, the participant was asked to assess the top 10 results. We decreased the number of assessed results to avoid burdening the participant with repeating the same task again. For more details on recommendation implementation, please see section 5.8

(v) After they finished the search task and checked the recommended content, they were asked to fill out questionnaires (see appendix E.1). Thus, they could give a score from 1, "Strongly Disagree" to 5, "Strongly Agree" for the level of enhancement in the precision of recommendations and the results to determine whether the results were relevant. After that, information analysis and obtained results will show whether our approach outperformed other approaches and to what extent the semantic network enrichment for the user profile and users' query enhanced the precision of the retrieved and recommended articles.

6.3.4 Comparative Systems

We considered five approaches for comparison in order to assess our hypotheses. These approaches are briefly described below:

- Google API[1,2] is the baseline and provides standard recommendations. It was adapted to give the user standard recommendations from our dataset.

- The recommender approach suggested by Mirizzi et al. [159] was designed to provide recommendations on movies (but, we adapted this approach to provide recommendations on bioinformatics articles) and extract semantic information from LOD and then use them to enhance the quality of the provided recommendations. Moreover, this approach considered the user profile to provide personalised services to each user based on his/her preferences. However, its user profile is not adaptable; it ignores the frequent change in the user preferences, and this may lead to inaccurate recommendations as a result of the change in the user preferences over time. Also, this approach only extracts semantic information from LOD; it does not successfully employ the extracted information by performing further inferences to discover more triples, which can enhance the quality of the recommendations. The Vector Space Model (VSM) [160] is used to handle semantic information that exists between LOD concepts.

  Furthermore, this approach considered other factors such as genre (i.e. comedy or tragedy) and weight for each property, represented by α ρ (i.e. a weight which is given to each property which represents its value regarding the user profile) and assigned by weight to represent the level of importance for a feature to the user. So, for the user who likes comedies, the property's weight will be high for such movies in this recommendation process. Since these factors do not fit with our approach, as our articles do not have these properties, we used a modified version in which these features were removed. Thus, equation 6.2 represents the new version equation that was considered

[1] https://cloud.google.com/prediction/docs

[2] https://developers.google.com/custom-search/?hl=en

to adapt this approach.

$$Profile(u) = \{m_j | u\ likes\ m_j\} \tag{6.1}$$

Equation: User profile formulation (Mirizzi et al. [159], page 9)

where u represents a user and m_j represents movies preferred by the user.

$$r(u\, m_i) = \frac{\sum Sim(m_j\, m_i)}{\mid Profile(u) \mid} \tag{6.2}$$

where r is the recommended item, m_j is movies stored (in our adapted system this represents preferred articles) in the user profile and m_i is movies (articles in our adapted system) that will be recommended to the user.

Furthermore, users' preferences were captured by following the same steps that applied into our main approach. However, we did not construct an ontological user profile for the participants who were assigned to this adapted approach. Since, Mirizzi's approach does not use an ontology to represent user preferences. Moreover, we did not apply the updating and deleting methods to the browsed content to make the user profile simulate the user profile of Mirizzi's approach. Furthermore, we considered that a participant prefers a particular piece of content when he or she spends more than 10 seconds reading that content (which is considered our threshold for deciding whether a user is interested in the content, for more detail please read section 5.3.1.3), since in the Mirizzi system, movies were added to the user's preferences when he/she presses "like" on a preferred movie. Thus, this represents a simulation of what was done in the Mirizzi approach. Then, participants were asked to choose four values to rate each result from the top 30 results (because it was difficult for users to rate all of the provided results), where "highly relevant" equals 4, "relevant" equals 3, "relevant to some extent" equals 2, and "not relevant at all" equals 1. The user then filled out the questionnaire to give a general indication of his/her opinion on all provided features.

- Our approach without the user profile to check the level of success that could be achieved with exploiting the sibling semantic relation without using the user profile. This will exploit the similarity between user query and different concepts in the inferred semantic network. This can be performed by direct matching between user's query and the inferred semantic network concepts. So, in a case of user's query is founded as concept in the inferred semantic network, then the recommended articles will be based on the first concept that has a sibling relation with the user query. But, if a user query does not exist in the inferred semantic network, then the cosine similarity will be calculated between the user's query and the inferred semantic network concepts. Then, recommended articles will be shown, based on the concept that has achieved the highest similarity score with the user query, as well as the first concept that has a sibling relation with the inferred semantic network concept that achieved the highest similarity score with the user query.

- Our approach without the semantic network will provide a user with recommendations based on his/her profile without exploiting the information that exists in the semantic network; this helps to provide more accurate and diverse recommendations.

- Our complete approach with all features (user profile and semantic network) will be compared with other approaches to see the level of success that can be achieved by this hypothesis in terms of the preciseness of the returned results.

6.3.5 Bioinformatics Recommender Services Evaluation

For evaluating our recommender approach, we conducted a user-centric evaluation to measure the level of satisfaction achieved by our approach in comparison to others in providing recommendations on the read content for people who specialise or are interested in bioinformatics. We followed the method of Knijnenburg et al.[193], which was used to evaluate the level of enhancement achieved when considering the semantic relations (i.e. siblings and semantic similarity relation in our second experiment section 6.4) in our recommender approach. Their method follows five steps for assessing the recommender system: (i) randomly select the set of participants, explain the purpose of the experiment and divide them into groups that contain an equal number of participants; (ii) each evaluated system should have a single group; and then (iii) participants should be asked to interact with five well-defined tasks (appendix E.2) as suggested in [217] and save the users' interaction in the database. Then (iv), ask the participants to fill out a questionnaire (appendix E.1) to determine their opinions regarding the different functionalities provided in the evaluated system. Finally (v), analyse the participants' data to measure the performance of each evaluated system.

Our evaluation method concentrated on the classification-accuracy and predictive accuracy metrics. Participants were asked to choose four values to rate each result from the top 30 results (because it was difficult for users to rate all of the provided results), where "highly relevant" equals 4, "relevant" equals 3, "relevant to some extent" equals 2, and "not relevant at all" equals 1. These ratings were applied on all participants from all comparative approaches. We only considered score 4 "highly relevant" and 3 "relevant" as a good recommendation and considered the other scores as bad recommendations.

Furthermore, a questionnaire covered the five compared approaches (i.e. our approach, our approach without the semantic network, our approach without the user profile, the literature approach [159] and the baseline approach) to assess the different features. It consists of eight statements to evaluate the level of success achieved by our approach. Half of the statements (1, 2,7 and 8) were taken from [218] and [193], as these statements are general to some extent and should exist in most of the recommender systems to measure the different factors (e.g. diversity, accuracy, novelty and satisfaction). The remaining statements (i.e. 3, 4, 5 and 6) were tailored to assess specific features in our recommender system. Thus, these statements have a scale from 1, "Strongly Disagree", to 5, "Strongly Agree". Therefore, through these eight statements, as well as calculating **Precision** at **N**, mean average precision (MAP) and mean absolute error (MAE) for each participant through all comparative approaches, we can assess the level of success in enhancing the preciseness of recommendations in our recommender approach in comparison with the other approaches.

6.3.6 Results

As discussed in the previous section, we ran this experiment with 30 participants and tested five comparative approaches (i.e. an approach using all functions, which is called BioRec_Full; an approach with a user profile only, which is called BioRec_SN; an approach featuring the semantic enrichment of the sibling's relations only, which is called BioRec_Profile; an approach integrating the literature [159], which is called Mirizzi; and the baseline, which is called Google). Several metrics compared the level of success in our approach with other comparative approaches from different perspectives. The following sub-sections will discuss all the applied metrics.

6.3.6.1 Precision at N Metric

This metric was applied to assess the level of success achieved by our approach (BioRec_Full) when compared to the other comparative approaches. Figure 6.1 shows the result of using **Precision** at **N** metric (BioRec_Full), which demonstrated a significant difference compared with the other approaches.

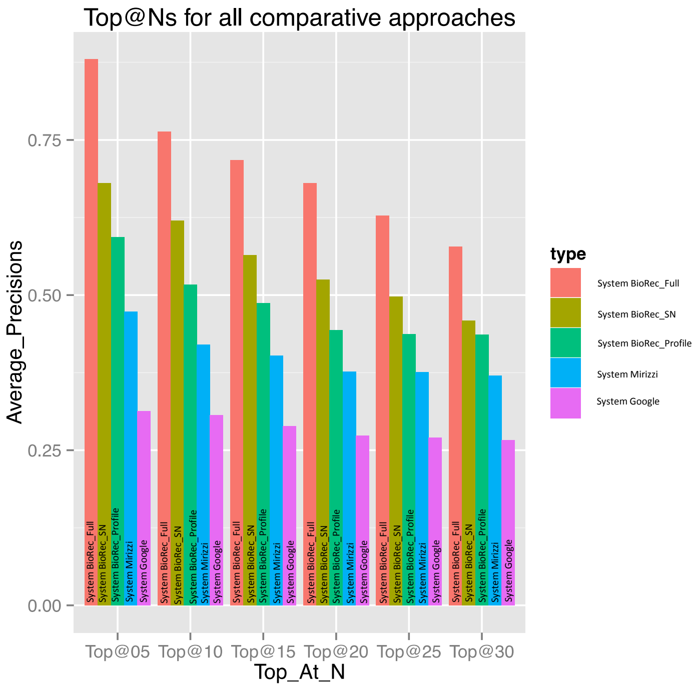


Figure 6.1: Average of Precision at N Metric for Bioinformatics Recommender Services.

So, system BioRec_Full outperformed the other comparative approaches in all calculated thresholds. On the other hand, a dramatic decrease in system BioRec_SN was registered, which reflected the importance of applying semantic network enrichment with the sibling's relation throughout the user profile. However, in the case of comparing system BioRec_SN with the remaining systems (BioRec_Profile, Mirizzi and Google), system BioRec_SN registered a significant difference in most thresholds (i.e. 5, 10, 15, 25) except the last threshold, which is 30. Regarding the latter, system BioRec_SN and BioRec_Profile registered close results, since top@30 reflects the overall rates and system BioRec_SN and BioRec_Profile have approximately the same number of preferred articles, but system BioRec_SN, with the user profile, registered a higher rate of scoring. System Mirizzi, on the other hand, had a lower score than the former systems (BioRec_Full, BioRec_SN and BioRec_Profile), since it did not support some functionalities that exist in these systems, such as dynamic adaptive ontological user profile and semantic enrichment with sibling's relation, which can enhance the precision of the provided recommendations. Finally, system Google, which registered the lowest score in this evaluation, where it provided standard recommendations that do not consider the ontological user profile or semantic network enrichment.

Moreover, a t-test [219] was applied to this evaluation, and it registered some significant results such as top@5 between BioRec_Full and BioRec_SN 0.01, where this score represents a significant difference on the t-test scale. Also, many different scores registered less than 0.01, such as: BioRec_Full & BioRec_Profile in top@15, which registered 0.0000426; BioRec_Full & Mirizzi in top@10, which registered 0.00000825; and BioRec_Full & Google in top@30, which registered 0.00131; etc. All these results represent a significant difference, which supports our approach and claims that a dynamic ontological profile and semantic network enrichment can enhance the preciseness of the provided recommendations.

6.3.6.2 Mean Average Precision Metric

Another metric was considered to ensure that all the results gained from the Precision@N metric were accurate and correct. For this purpose, the Mean Average Precision (MAP) was used. It registered similar results, demonstrating that our results are similar across more than one metric. Figure 6.2 represents the results of MAP for the five compared systems.

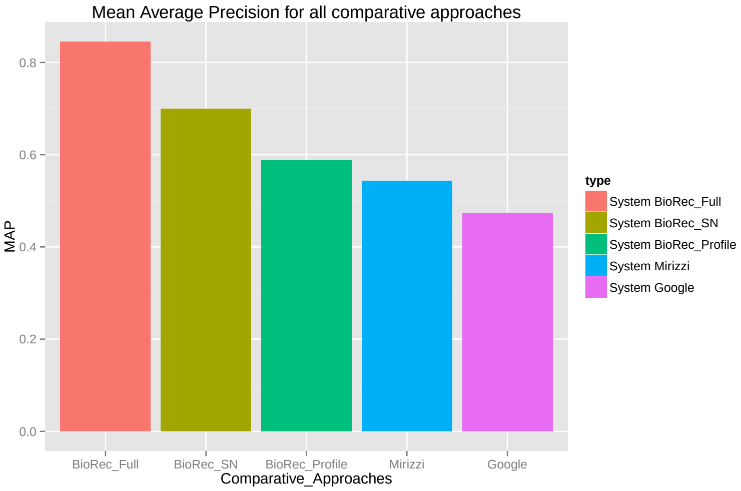


Figure 6.2: Mean Average Precision Metric for Bioinformatics Recommender Services.

Thus, figure 6.2 shows that system BioRec_Full outperformed the other comparative approaches by a significant score. This reflects the importance of an adaptive ontological user profile as well as semantic network enrichments with sibling relation over multiple bioinformatics resources. System BioRec_SN, which represents the second system in comparison with the remaining approaches (BioRec _Profile, Mirizzi and Google), shows how a dynamic ontological profile enhances the precision of the recommendations provided. Moreover, systems BioRec_Profile and Mirizzi registered close results, but system BioRec_Profile outperformed both, thereby reflecting the importance of semantic network enrichment and its ability to enhance the preciseness of the recommendations provided. Finally, system Google had the lowest results because it provided standard recommendations rather than personalised ones.

Moreover, the t-test [219] metric was applied to the results gained from the MAP metric, and it registered significant results. For instance, the t-test between system BioRec_Full and system BioRec_SN registered a score of 0.010, which was considered a significant difference on the t-test scale; the score between system BioRec_Full and system BioRec_Profile registered a score of 0.00000726; the score between BioRec_Full and Mirizzi was 0.0000162; and, finally, the score between system BioRec_Full and system Google was 0.000000896. Such results suggest that our approach enhances the precision of the recommendations of academic articles in the field of bioinformatics.

Table 6.1: MAP for All Assigned Tasks in Sibling Enrichment Comparison

Systems/Tasks	Task 1	Task 2	Task 3	Task 4	Task 5
BioRec_Full	0.8594	0.7733	0.8054	0.8917	0.8946
BioRec_SN	0.5890	0.7017	0.6620	**0.7619**	**0.7833**
BioRec_Profile	**0.6044**	0.5797	0.6093	**0.6459**	**0.4977**
Mirizzi	**0.6048**	0.5326	**0.5305**	**0.5751**	**0.4743**
Google	**0.4674**	**0.4419**	**0.4077**	**0.5735**	**0.4770**

Furthermore, a Mann-Whitney U-Test [219] was applied which is more accurate on calculating of significant score than t-test, and it registered some significant results such as between BioRec_Full and BioRec_Profile; and BioRec_Full and Mirizzi; and BioRec_Full and Google where all of the former registered Z score equals 2.5067 and p-value 0.00604 and this considered as significant when $p \leq 0.01$; and U score equals 0 and the critical value of U equals 1 at $p \leq 0.01$ where this means it is significant based on results[1]. Table[2] 6.2 discusses the significant results in each assigned task in more detail.

Table 6.2: MAP for All Assigned Tasks in Sibling Enrichment Comparison for Mann-Whitney U-Test significant Results

Systems/Tasks	Task 1	Task 2	Task 3	Task 4	Task 5
BioRec_Full	0.8594	0.7733	0.8054	0.8917	0.8946
BioRec_SN	0.5890	0.7017	0.6620	0.7619	0.7833
BioRec_Profile	0.6044	0.5797	0.6093	**0.6459**	**0.4977**
Mirizzi	0.6048	0.5326	**0.5305**	**0.5751**	**0.4743**
Google	**0.4674**	0.4419	**0.4077**	0.5735	**0.4770**

[1] http://www.socscistatistics.com/tests/mannwhitney/

[2] All records in bold are representing significant scores.

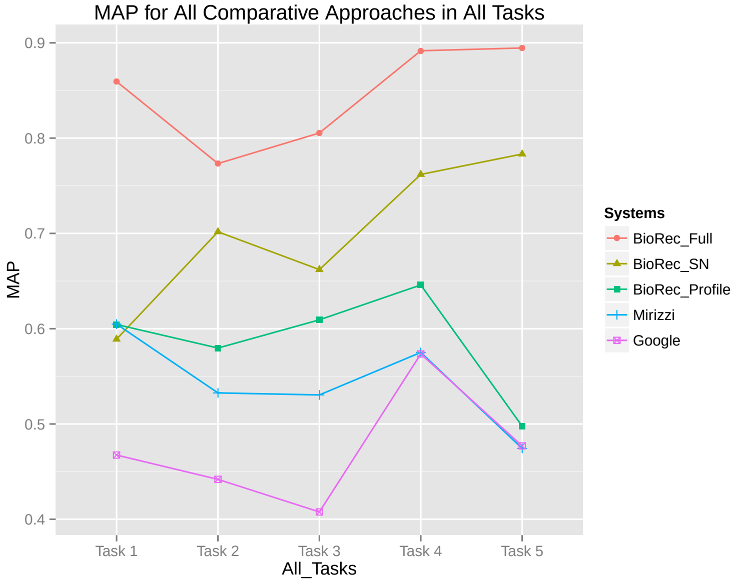


Figure 6.3: MAP for All Tasks in Each Approach.

Table[1] 6.1 shows the MAP scores that were satisfied in each system regarding each assigned task. These tasks were designed to test different functions provided by our recommender system, starting from a general task and gradually moving to a complex one. Figure 6.3 demonstrates an overview of the MAP scores which are satisfactory in all assigned tasks. As shown in **task 1** appendix E.2, the best approach was BioRec_Full, which had outperformed the other approaches in providing recommendations on general topics in bioinformatics. The reason behind this was the exploited sibling relation which supports this approach to provide recommendations semantically connected to the general topics that are preferred by the bioinformatician. Also, the adaptive ontological user profile was supporting this task, which helps our method to provide more accurate recommendations. Then, system (BioRec_Profile and Mirizzi) got very close results and outperformed the BioRec_SN. This related to the importance of applying semantics to make more accurate recommendations in this task. BioRec_SN depends on user profile only, so the recommendations will be concentrated on the preferred topics that are stored in the user profile. Thus, if specialists did not use to read general topics, recom-

[1] All records in bold are representing significant scores.

mendations will not be accurate as they expected. Google, the standard method, was the worst approach for this type of recommendations.

Task 2 saw a decrease in all comparative approaches except BioRec_SN. BioRec _SN is depending on the user profile; the specialist in this approach is used to use a specific tool in bioinformatics and the user profile may have the tool name that specialist prefer. Thus, it provides recommendations based on the specific tool (or brand) that the user likes. However, the other approaches (BioRec_Full, BioRec_Profile and Mirizzi) did poorly in this task, with BioRec_Full in sharp decline. Semantics appear weak in this task, however, with this decrease BioRec_Full still outperformed the other comparative approaches. We believe Google provides personalised recommendations, but it does not employ any semantic method that exploits semantic relations and hidden association between multiple resources and its method is still weak and unable to provide accurate results.

Task 3 shows an almost opposite reaction to task 2 where all approaches that were suffering from decrease in task 2, increased their performance and vice versa, except Google, which still decreased. This confirmed the role that could be played by semantics in enhancing recommendations provided in BioRec_Full, BioRec_Profile and Mirizzi. Those approaches that depend only on the user profile did not work well here. Because, recommendations that discuss some languages that used in bioinformatics need to apply semantics more than user preferences, since this task is a bit tricky and considering semantic relations (such as sibling) to provide recommendations can address such problem. In addition, **task 4** led to an increase in all because of the compatibility between the semantics and using the user profile, but Mirizzi and Google achieved very close results. This shows the weakness of the user profile and semantics used in Mirizzi which make its performance not far from standard systems such as Google. Finally, for **task 5** BioRec_Full and BioRec_SN increased their performance while it sharply decreased in the other approaches. This could be related to the accuracy of the ontological user profile that is used in these approaches which help them to enhance their accuracy. Moreover, BioRec_Full has outperformed the other approaches even BioRec_SN which also used an ontological user profile. This accomplishment was achieved as a result of using the sibling relation in the BioRec_Full approach.

Moreover, all tasks that have been provided to the participants to assess different functionalities of each approach will be discussed in more detail to clarify the reasons behind the strength and weakness of each comparative approach. For instance, **task 1** (appendix E.2) was designed to test the utilisation of system BioRec_Full (which was equipped with the semantic relation (i.e. sibling relation) as well as an automatic adaptive ontological user profile) and compared with the other systems (BioRec_SN, BioRec_Profile, Mirizzi and Google) in order to provide recommendations on articles that discuss general ideas about bioinformatics such as definitions, histories etc. So, the bioinformatician can refresh his/her knowledge with the general information about bioinformatics.

Figure 6.4 demonstrates that system BioRec_Full outperformed the other systems. Moreover, it shows that system BioRec_SN had a dramatic decrease in comparison with BioRec_Full and also performed less well than systems BioRec_Profile and Mirizzi, but it was better than system Google. This suggests that only using an automatic adaptive ontological user profile is not effective in providing recommendations about general topics; however, it is still sufficient in comparison with system Google, which provided standard recommendations. Both systems BioRec_Profile and Mirizzi achieved very close scores in MAP, and this can indicate two main conclusions: (i) semantic enrichment demonstrates an important role in enhancing recommendations without even using a user profile. (ii) The user profile used in system Mirizzi is still weak because it does not use ontologies to represent the user profile and does not apply a method that can keep the user preferences updated. Also, it does not successfully exploit semantic relations and employ them in order to enhance recommendations. However, both systems have good performance in comparison with system Google, which represents the weakest system in this comparison, as it provides standard recommendations.

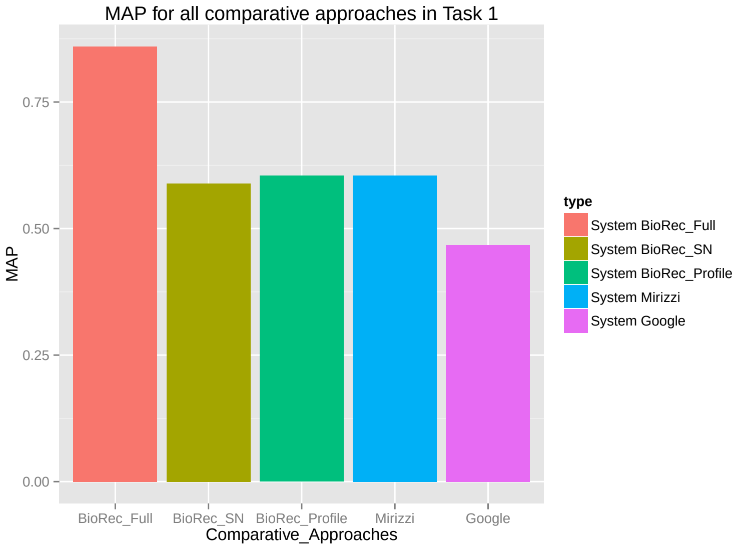


Figure 6.4: Mean Average Precision Metric for Task 1.

Furthermore, concerning **task 2** (appendix E.2) and **task 3** (appendix E.2), the former was designed to recommend the user with articles that discussed tools, which are useful for the bioinformatician such as the tool for DNA sequencing, alignment or annotation. The latter was designed to recommend the user with articles that discuss programming languages that were used in bioinformatics such as Matlab, Perl, Biojava, etc. Figures 6.5 and 6.6 clearly show that the systems' utilisations gradually decrease from systems BioRec_Full until they reach the weakest system in both figures, which is Google. These results are close to each other because they assess two functions that could be similar to each other; for example, the tools used in bioinformatics could be used on one of the programming languages mentioned before. Thus, the performance of each system has registered a similar score in MAP for both tasks. So, as shown in both figures (6.5 and 6.6), system BioRec_Full exceeded the other compared systems, and this success can reflect the importance of both ontological user profile and semantic enrichment to provide better recommendations. Then, system BioRec_SN in figure 6.5 is close to BioRec_Full, which indicates that the user profile has good performance in such tasks, and the difference between BioRec_Full and BioRec_SN in this figure reflected the role that the sibling relation played to enhance recommendations.

However, system BioRec_SN in figure 6.6 had a dramatic decrease in comparison with BioRec_Full, which may only reflect the weakness of considering the user profile only in recommending articles that discuss programming languages used in bioinformatics. Since some of the participants may not be experts with these programming languages and using semantics in such a case becomes an essential need. This can be confirmed by the use of system BioRec_Profile in both figures, where system BioRec_Profile used only semantics, and, as shown in both figures (6.5 and 6.6) in the first figure 6.5 (task 2), it had bad performance in comparison with its performance in 6.6 (task 3), which depended on the semantics more than the user profile.

Both systems of BioRec_Profile in these tasks had a good performance in comparison with the literature system (Mirizzi) and general system (Google). System Mirizzi satisfied similar results in both tasks, which reflected the importance of using the user profile in Mirizzi's method and applying semantics to provide recommendation. However, they are still insufficient compared with applying (semantics and automatic ontological profile) BioRec_Full, (user profile only) BioRec_SN or (semantics only) BioRec_Profile. Yet they outperformed the Google system, which provides standard recommendations. In both figures 6.5 and 6.6, the Google system has not considered ontological user profile or semantic relations such as sibling, and for this reason it had the lowest score in this comparison.

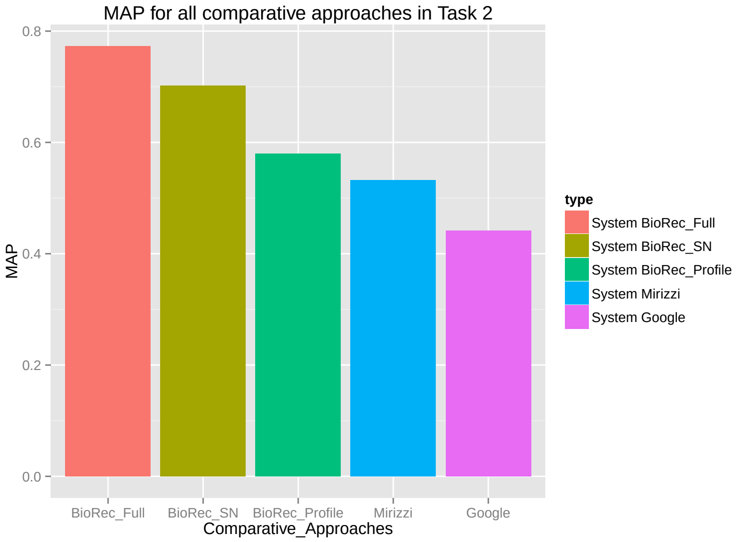


Figure 6.5: Mean Average Precision Metric for Task 2.

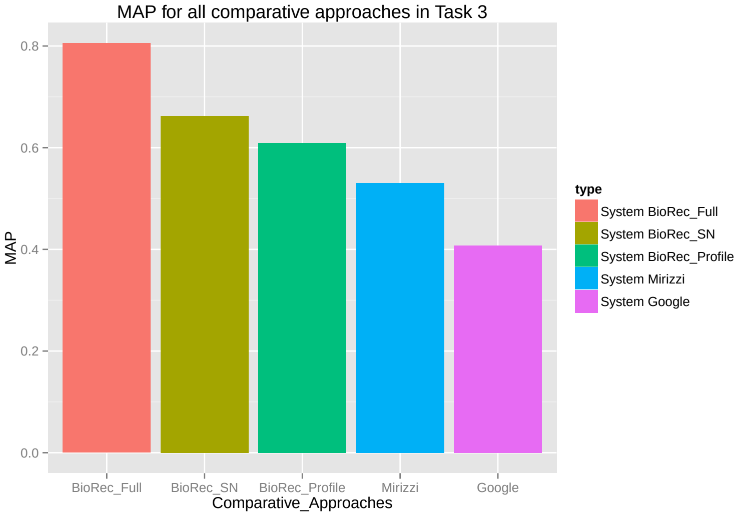


Figure 6.6: Mean Average Precision Metric for Task 3.

Additionally, **task 4** (appendix E.2) was designed to recommend articles to the user that discuss bioinformatics ontologies such as GO or PO, and this task was more complex than the previous ones. Since it should recommend some articles to each user that discuss ontologies, which are preferred by the user or any relevant articles based on the semantic relation exploited in this experiment. Figure 6.7 demonstrates that system BioRec_Full outperformed the other compared systems. Then, gradually system BioRec_SN and BioRec_Profile, where the former in this task had a moderate difference in comparison with system BioRec_Full, reflect the effectiveness of the user profile which is able to recommend the user with the most relevant articles to his/her profile. The latter had a moderate difference in comparison with system BioRec_SN because the sibling relation had a sufficient effort to enhance the recommendations. Moreover, systems Mirizzi and Google were equal in their use in this task, and this may reflect the weakness of the system Mirizzi with a task classified as a complex task. On the other hand, it can reflect that our baseline was quite good, even with a complex task such as this one.

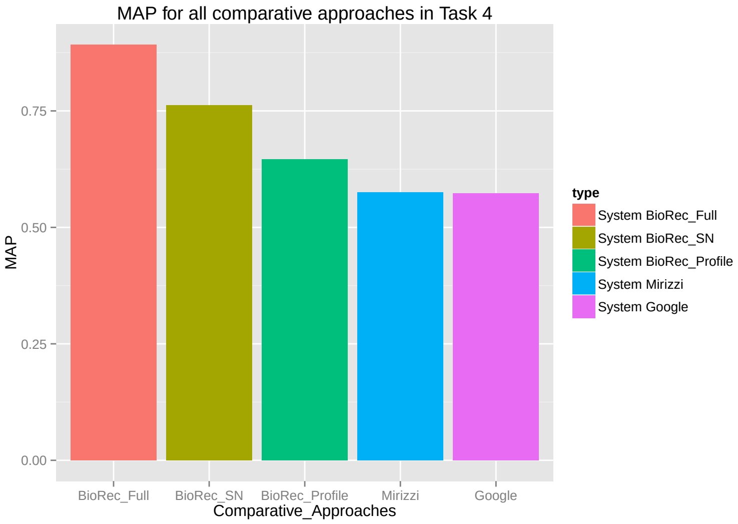


Figure 6.7: Mean Average Precision Metric for Task 4.

Task 5 (appendix E.2) was designed to recommend the bioinformatician with articles that mentioned some bioinformatics journals such as BMC, Computational Molecular Biology, etc. This task may help him/her to broaden his/her horizons and become aware of the most important journals in the field. Moreover, it can show the level of accuracy that our recommender approach has achieved in using both semantic enrichment (i.e. exploiting sibling relation) and user profile to provide better recommendations. As figure 6.8 demonstrated, the BioRec_Full system outperformed the other compared systems. Also, it showed that there was not much difference between BioRec_Full and BioRec_SN in comparison with the other approaches. This is confirmed the role that was added by using the ontological user profile and the difference between BioRec_Full and BioRec_SN, and which showed that BioRec_Full has overcome BioRec_SN as a result of considering semantic enrichment to enhance the accuracy of recommendations. System BioRec_Profile had a dramatic decrease in comparison with BioRec_Full and BioRec_SN, and this emphasises that only applying semantics without taking the user profile into account can weaken the quality of the provided recommendations. Again, system Mirizzi and Google had equal utilisation, and this reflects that system Mirizzi did not consider an ontological adaptive user profile and did not properly employ the semantic relation. Moreover, this task can emphasise that system Google, as a baseline, was able to support users with valuable recommendations. However, it was still weak in comparison with the other approaches (BioRec_Full, BioRec_SN and BioRec_Profile). This weakness appeared as a result of using standard method to provide recommendations and ignoring the semantic relation, such as sibling, to enhance the accuracy of the recommendations.

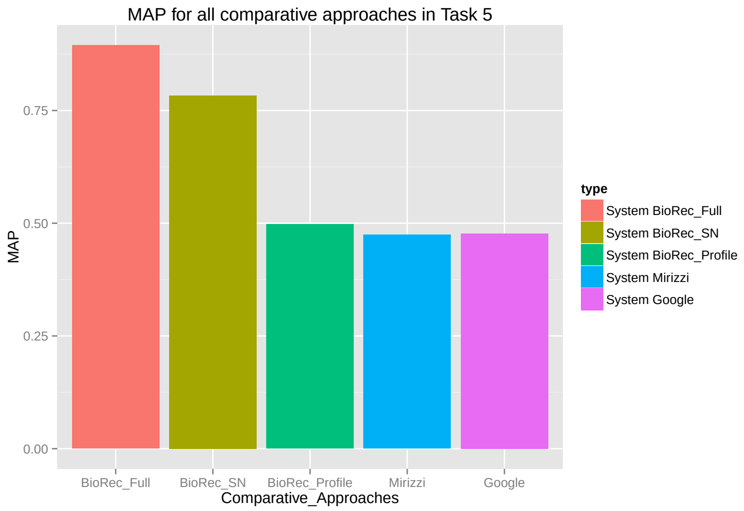


Figure 6.8: Mean Average Precision metric for Task 5.

Finally, another comparison was taken for all comparative approaches for the same tasks. However, this time users were asked to assess the recommendations based on specific interests they chose from their profiles. They were asked to assess only the top 10 results, and this threshold was chosen so as not to burden the users with assessing the same task again. This evaluation was completed by considering two evaluation metrics, which were considered in the previous test (i.e. Top@N and MAP) to assess the level of success achieved when selecting specific interests to have recommendations on. Figure 6.9 and 6.10 show the result of systems' use regarding each assigned task.

As shown in figure 6.9, system BioRec_Full outperformed all comparative thresholds, and this can be illustrated by the increased accuracy of the provided recommendations when both semantic relations (i.e. sibling) as well as adaptive ontological user profile were used. Surprisingly, system BioRec_SN in the same figure had a dramatic drop in the Top@5 threshold. This can be explained by the weakness of the recommender system when the preferences were narrowed down into a specific interest, and so only considering the user profile for providing recommendations was not quite sufficient. Also, system BioRec_Profile, which only uses semantic enrichment, outperformed this system even though it was not equipped with method that provide recommendations based on a specific interest, and this showed the weakness of considering the user profile in such an approach.

Moreover, system BioRec_SN at Top@10 increased the accuracy of the provided recommendations, where such an enhancement can be explained as a result of increasing the number of assessed results. So, the users can find their articles of interest more accurately than the previous threshold. System BioRec_Profile, in this assessment, did quite well in comparison to Mirizzi and Google, and this can reflect the effectiveness of the semantic enrichment to the user's query, which may have sufficient influence on the provided recommendations. Although, system BioRec_Profile was not supported with method that allows it to provide recommendations based on determined interest. System Mirizzi satisfied similar results in both thresholds, which may reflect the effectiveness of applying the user profile (i.e. not adaptive) and can enhance the recommendations when compared with the other system (i.e. Google). However, this system is still weak when it was compared with the other three systems (BioRec_Full, BioRec_SN and BioRec_Profile) when the average of both thresholds (top@5 and top@10) were considered, and this shows that system Mirizzi was not adequate, even when considering the user profile and employing semantics to provide recommendations. Also, Mirizzi was not able to provide recommendations based on a specific interest; it provided recommendations based on all user's preferences stored in the user profile. Then, system Google was the weakest in both thresholds because it provided standard recommendations, did not exploit semantic relations to provide recommendations, and was not supported with a method to provide recommendations based on specific interests.

Furthermore, the results shown in figure 6.10 were not far from the Top@N metric. System BioRec_Full outperformed the other systems. Then, the performance gradually decreased when moving to another system until reaching system Google. This supports our hypothesis that considering both semantic relations (i.e. sibling) and the adaptive ontological user profile can contribute to enhancing the accuracy of the provided recommendations in all user's preferences or in a specific interest.

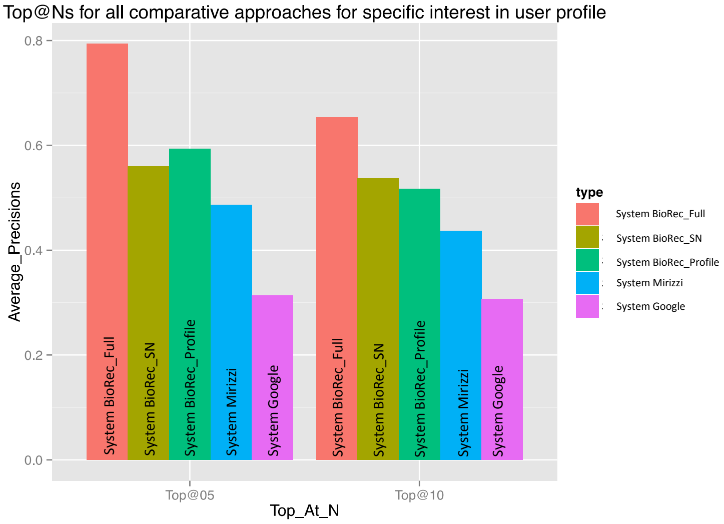


Figure 6.9: Top@N for All Comparative Approaches in Specific Interest.

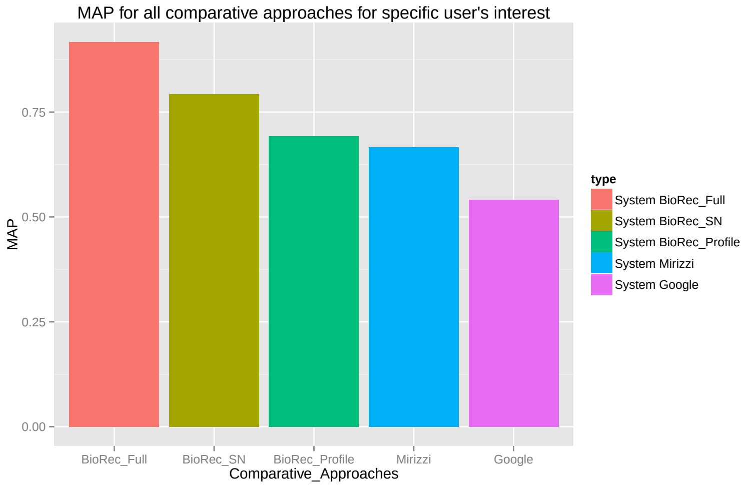


Figure 6.10: MAP for All Comparative Approaches in Specific Interest.

6.3.6.3 Mean Absolute Error Metric

Another metric mentioned in section 3.5 and used for predictive assessment was applied into our comparative approaches to check which one has achieved the lowest score of error in predicting recommendations to the user. As seen in Figure 6.11, BioRec_Full has outperformed the other approaches. Since it had the lowest MAE score, BioRec_Full seems to draw this accuracy from both semantic enrichment with sibling relation and adaptive ontological user profile. The second best performance of system BioRec_SN, which achieved the second lowest MAE score, also shows the importance of user preferences and how adaptive ontological user profile can enhance the accuracy of the provided recommendation and make the error lower than other three comparative approaches (i.e. BioRec_Profile, Mirizzi and Google). System BioRec_Profile, equipped with semantic enrichment with sibling relation, was almost as good as BioRec_SN and close in score to Mirizzi's system, where this can lead to conclude that semantic enrichment has worked well without user profile support. System Mirizzi is still weak in exploiting semantic relation to enhance the accuracy of predictive recommendations. Also, its user profile (which is not an automated ontological user profile) is still weak in predictions. Finally, the Google system had the highest MAE score and did worst. This can indicate that the error chance is a bit higher when using standard method to predict recommendations. Also, it was not supported by either semantic enrichment or adaptive ontological user profile. It only has a regular profile which is not accurate enough for this case.

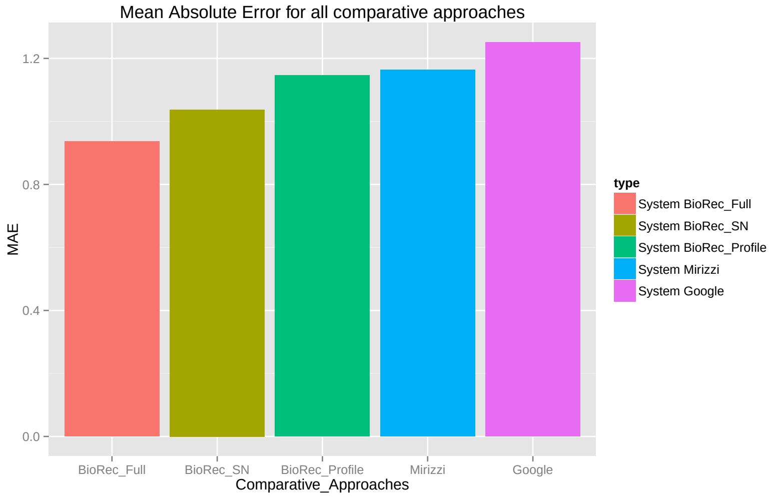


Figure 6.11: Mean Absolute Error for All Comparative Approaches.

Moreover, the MAE metric was used to assess the accuracy of our approach compared with other comparative approaches (i.e. BioRec_SN, BioRec_Profile, Mirizzi and Google) by asking users to select a specific interest of their preferences for recommendations. This time users were asked to assess the top 10 results only, for the same reason mentioned when we applied the other metrics (MAP and Precision@N) to the same task (getting recommendations based on specific interests they chose from their profiles).

Figure 6.12, points out that system BioRec_Full outperformed the other comparative approaches as a result of using both semantic enrichment (with sibling relation) and adaptive user profile help to enhance the level of accuracy in predicting new articles to the users. After that, BioRec_SN had a surprising, dramatic increase in the level of error in comparison with system BioRec_Full and a moderate increase compared with BioRec_Profile, despite that BioRec_Profile was not equipped with method to provide recommendations based on a specific interest. The reason behind outperforming both (BioRec_Full and BioRec_Profile) is the combination of semantic enrichment with sibling relation rather than only the use of adaptive ontological user profile. Then, Mirizzi showed a dramatic increase compared with BioRec_Profile, even though the feature of determining a specific interest to get recommendations on it was not applied in both systems (BioRec_Profile and Mirizzi). This shows the importance of exploiting semantic relations correctly and then exploit them to enhance the recommendations. Moreover, the Mirizzi system did quite well compared with our baseline (i.e. Google), which was not equipped with any semantic enrichment. Finally, Google was the worst system in this comparison, because it was only using standard method to provide recommendations which is not as accurate compared with the other methods. Also, it was not supported with the method that allows specialist to select a specific interest to receive recommendations based on the selected one and its recommendation was based on all preferences.

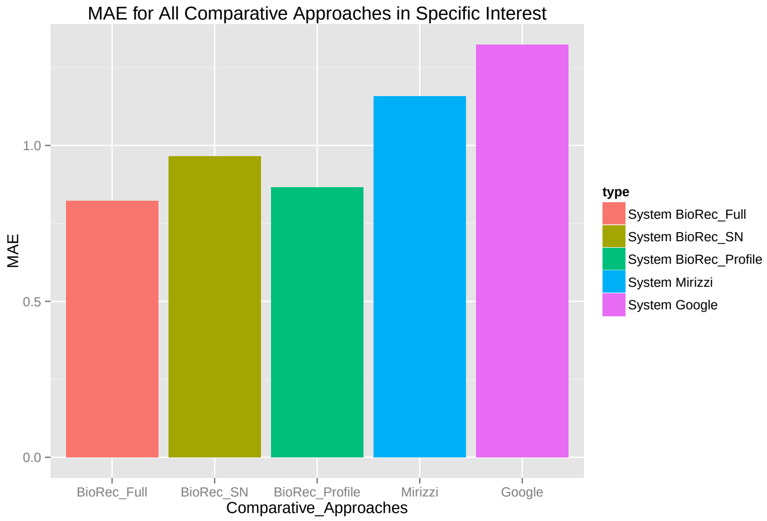


Figure 6.12: MAE for All Comparative Approaches in Specific Interest.

Table 6.3 shows the average for the questionnaire statements selected. As shown in table 6.3, participants in group BioRec_Full were satisfied with most of the functionalities provided by system BioRec_Full. System BioRec_SN's participants were the second most satisfied. System BioRec_Profile's and system Mirizzi's results were close or similar to each other in some statements, such as statement 2, statement 5 and statement 6. Finally, system Google registered the lowest score,

Table 6.3: Questionnaire Evaluation for Bioinformatics Recommender Service (Stm: Statement)

Systems	Stm 1	Stm 2	Stm 3	Stm 4	Stm 5	Stm 6	Stm 7	Stm 8
BioRec_Full	4.8333	4.6667	4.5	4.8333	4.5	5	4.8333	4.6667
BioRec_SN	4.1667	3.8333	3.8333	3.5	3.3333	4	4	3.5
BioRec_Profile	2.6667	3.5	2.8333	3.6667	2.5	1	3.1667	3
Mirizzi	3.1667	3.5	3	3	2.5	1	2.8333	2.8333
Google	1.5	3	2	2.1667	1.8333	1	2.3333	2.3333

and this reflects the fact that participants in this system only received generic recommendations. This questionnaire represents an overview for the quality of the recommendations provided and supports all evaluation methods (i.e. Precision@N, MAP and MAE) applied to the comparative approaches.

6.3.7 Experiment Summary

To this end, and based on the analysed data and gained results. We can conclude that our approach is novel with the exploited relation which was sibling relation. Since, exploiting this relation demonstrated its ability to support specialist search in specific domains such as bioinformatics and enhance the accuracy of the provided recommendations. This relation was exploited by our recommender system which is equipped with adaptive ontological user profile, which collects user preferences automatically and tailors recommendations to each specialist based on his/her preferences. This method has been compared with several methods such as our approach from different perspectives, including our own recommender without considering user profile and without considering the semantic relation; and general method and method from literature that exploits semantics from different resources [159]. The Mirizzi method exploits semantics between different resources, but is still weak in this exploitation since it does not perform further inferencing from the extracted information. Such inference may lead it to discover new relations and information that could help in enhancing the accuracy of the provided recommendations. Moreover, the user profile which is used in this method was weak compared with our considered user profile. There are couple of reasons for such weakness. i) The profile used in [159] was not ontological. The ontological user profile can help them discover new data that could enrich the user profile with new relations and information, which support the user profile and enhance the precision of the provided recommendations. ii) Their profile was not fully automated, because it concerns adding new preferences but ignores updating and deleting them. This makes their recommendation method unable to provide up-to-date recommendations, since it is not able to respond to the frequent changes made in the specialist preferences.

6.4 Experiment 2: User Centric Evaluation for Semantic Similarity Method

The aim of this experiment is very similar to the previous experiment where both try to enhance the accuracy of the provided recommended content (i.e. article). However, there are some differences between them which can be listed as i) type of semantic exploited relation (semantic similarity a new discovered relation). ii) Some of the comparative approaches (the participants in this experiment were given access to a range of systems (Google API as a baseline, our recommender approach that exploits semantic similarity developed method, our recommender approach that exploits sibling relation and system from the literature [159])). iii) The information-seeking tasks (appendix F.2) to be carried out by participants. iv) The way of collecting participants' data. So, the data were collected in three stages:

(i) Collecting data to formulate an ontological user profile as part of the first session.

(ii) Collecting data that represents user interactions (clicks, rates, etc.) with the systems while performing the eight assigned tasks during the second session.

(iii) Collecting the questionnaires (appendix F.1), which reflect the level of satisfaction that participants have with the provided recommendations and with regards to the used recommender system.

All data will be stored anonymously in log files and connected with each participant's user ID in order to link between each user and the tested systems. These data will be used to analyse the utilisation of each system and draw conclusions on the effectiveness of the developed methods.

6.4.1 Experiment Goal

There are four hypotheses that represent the main goal of this experiment:

- H1: Reasoning through multiple resources and extracting semantic relations (such as semantic similarity) and hidden associations. Then exploit this relation to enrich the ontological user profile and user query, in turn enhancing the recommendations and improve the retrieved results.
- H2: An ontological user profile which has been enriched (for further details about user profile enrichment with a discovered relation please see section 5.3.3) with the semantic similarity relation can support our recommendation method to provide more accurate recommendations.
- H3: The semantic similarity method supported with a method that allows a specialist to narrow down his/her topics in the user profile into a single topic, to get recommendations exclusively on that selected one, can enhance the accuracy of the provided recommendations on that selected topic.
- H4: The semantic similarity relation can enhance the accuracy of the provided recommendations, when the specialist did not submit any query to the system.

Thus, this experiment is meant to assess our recommender approach in comparison with other recommender approaches in the aforementioned hypotheses. Moreover, it will assess which of our discovered relations (i.e. siblings or semantic similarity) can add more enhancement in the accuracy of the provided recommendations. It will compare them with general approach, such as Google, and with other approaches from the literature [159].

6.4.2 Experiment Participants

Participants were employees from the national biotechnology centre at King Abdulaziz City for Science and Technology in Saudi Arabia, which specialised in different branches of the biological studies such as medical microbiology, molecular medicine, biotechnology, medical biochemistry, biological science and bioinformatics. All participants had good experience in bioinformatics where this centre concentrates its research in the field of bioinformatics. Thus, they can judge relevant content and the accuracy of the returned recommendations against their queries and preferences in the user profile. The number of subjective participants in the study was 24[1], and all participants were males.

6.4.3 Experiment Process

The experiment was run through several steps described as follows[2]:

(i) The user profile was first constructed for each specialist as discussed in section 5.3.2

(ii) After collecting data, participants were divided into four groups and assigned eight tasks (appendix F.2) to perform using one of the recommender systems.

(iii) The recommendations process in this experiment will be following the same procedure that has been performed in the previous experiment, but with some minor differences:

  - The specialist will read the assigned task (appendix F.2). His submitted query will be enriched with the semantic similar concept to the most similar concept stored in the user profile that matches his query.
  - The retrieved results from the Lucene search engine will be organised based on similarity between the specialist's query and his stored preferences.

[1] We have selected only this small number of subjects because conducting user-centred evaluation in the field of recommendation is known to be difficult and expensive [189].

[2] In this experiment, we have used the Ge and Qiu [208] method with equation 4.2 to calculate the weight between nodes while we formulate the semantic network, instead of using the method suggested by Stuckenschmidt and Schlicht [216]. This was based on several tests conducted between the two methods; we have found that the Ge and Qiu [208] method is more accurate to calculate the weight of the concept and helps achieve better semantic similarity scores.

- The list of top 30 recommendations in the link called "See Recommendations" will be related to the concept that has the highest semantic similarity score to preferences. This will be organised based on relevancy to his query.

Note: In task 6 (appendix F.2), which checks the accuracy of the system in providing recommendations without asking for specialist's query, only the last step will be executed. The specialist will click on the link, so the result will be the top 30 articles recommendations. The link contains articles that are related to the concept that has the highest semantic similarity score to his preferences. For more details about the recommendation implementation, please see section 5.8

(iv) After participants finish the search tasks and check the recommended content, they were asked to fill out questionnaires (appendix F.1). Thus, they can give a score from 1, "Strongly Disagree" to 5, "Strongly Agree" for the level of enhancement in the precision of recommendations and results to indicate whether the results are relevant. We will then examine whether our approach outperformed the other approaches. Also, this can let us know to what extent the semantic similarity relation exploitation can enhance the accuracy precision of the recommended articles.

6.4.4 Comparative Systems

We considered four systems for comparison in order to assess our hypotheses. These systems are briefly described as follows:

- Google API is the baseline and provides standard recommendations; it has been adapted to give the user standard recommendations from our dataset.
- The recommender approach suggested by Mirizzi et al. [159] was a comparative approach discussed in the literature. Section 6.3.4 provides more details about this approach.
- Our approach exploiting the sibling relation shows the level of success of our hypothesis when considering only this relation to enrich the user profile

(for further details about user profile enrichment with a discovered relation please see section 5.3.3).

- Our approach exploiting semantic similarity method to show the level of enhancement that can be acquired when considering such relation to enrich the user profile as well as the user query.

6.4.5 Semantic Similarity Method Evaluation

This method will follow the same evaluation steps applied in the previous experiment. The difference between them is only on the assigned tasks (appendix F.2) and the given questionnaire (appendix F.1). So, the questionnaire statements (1, 2, 9, and 10) are taken from [218] and [193] designed to measure general factors such as accuracy and diversity. However, the remaining six statements (3 to 8) are designed to assess the specific functions that exist in the recommender approach we used.

6.4.6 Results

This section presents the result of 10 statements that were given to the participants as an exit questionnaire (appendix F.1); each statement examines different functionalities of our recommender approach. Moreover, the evaluation results of the eight assigned tasks (appendix F.2), conducted on the four compared systems, are also presented here, namely: system BioRec_Sim, our recommender approach that uses the semantic similarity method; system BioRec_Sib, our recommender system that uses the sibling method; system Mirizzi, a system from the literature [159]; and system Google, which represents our baseline. As shown in table 6.4, system BioRec_Sim outperformed the other systems. This accomplishment reflects the level of user satisfaction that the approach achieved. Although system BioRec_Sib registered lower scores than system BioRec_Sim, it was still considered acceptable by users in various functionalities. System Mirizzi achieved average results in the general question; however, it had some weaknesses in specific functionalities. Finally, system Google, which was the weakest approach in the comparison, may reflect the importance of applying semantics and personalisation to enhance the

Table 6.4: Questionnaire Statements Evaluation of the Bioinformatics Recommender Service for Semantic Similarity Method (S: Statement)

Systems	S 1	S 2	S 3	S 4	S 5	S 6	S 7	S 8	S 9	S 10
BioRec_Sim	5.0	4.83	3.5	4.83	4.66	4.3333	4.66	4.83	5.0	4.83
BioRec_Sib	4.0	3.5	3.33	3.83	3.83	4.16	3.83	2.83	3.66	3.5
Mirizzi	2.66	2.66	3.0	2.33	3.0	2.16	1.16	1.16	2.5	2.5
Google	1.66	2.16	2.5	2.0	2.33	1.16	1.16	1.16	2.0	1.83

provided recommendations. This can show that our approach, which exploits sibling or semantic similarity relation, is a new approach that employs these inferred relations to support the specialist in bioinformatics domain and enhance the accuracy of the provided results. Furthermore the results of information analysis taken from different users' interactions with our prototype system can help us to measure the level of enhancement achieved by exploiting such relations. The following sections will discuss the evaluation metrics and their results that used to measure the level of enhancement, that could achieved when exploiting the semantic similarity relation in our prototype recommender system.

6.4.6.1 Precision at N Metric

We have undertaken this step to examine our suggested method in different thresholds. Figure 6.13, demonstrates that BioRec_Sim has outperformed the other comparative approaches in all thresholds (i.e. precision @5, @10, @15, @20, @25 and @30) in all assigned tasks (appendix F.2). This reflects that our hypothesis that exploiting semantic similarity relation and ontological user profile can enhance the accuracy of the provided recommendations. In addition, it shows how exploiting this inferred relation can support specialist researchers in specific domain with better results. Furthermore, it distinguishes our approach from other comparative approaches and confirms its novelty in employing such relation to enhance the quality of the provided recommendations. After that, BioRec_Sib had a dramatic drop compared with BioRec_Sim Top@5, @10, @15, and @20, although both relations can enhance the provided recommendations. However, this can strengthen our suggestion that semantic similarity with ontological user profile has more enhancements than sibling relation, especially in the first four thresholds. Thus, in

Top@25 and @30, we can observe that the difference between the two methods is not as dramatic. This is because BioRec_Sim focused on the top results, usually very important to most researchers, then its performance decreased, however, it is still the best approach in this comparison. After that, system Mirizzi had a significant difference compared with BioRec_Sim and BioRec_Sib in all thresholds and assigned tasks. This confirmed that even though the Mirizzi system uses semantics and user profile, but it is still weaker compared with our exploited relations (semantic similarity and sibling). Hence our used relations as well as the ontological user profile can add more enhancements on the provided recommendations than the Mirizzi system. Finally, the Google system was the weakest system in all thresholds, however, it was very close to the Mirizzi system. This can lead to conclude that the enhancement that the Mirizzi system did is not very far from the standard way which is used by Google to provide recommendations.

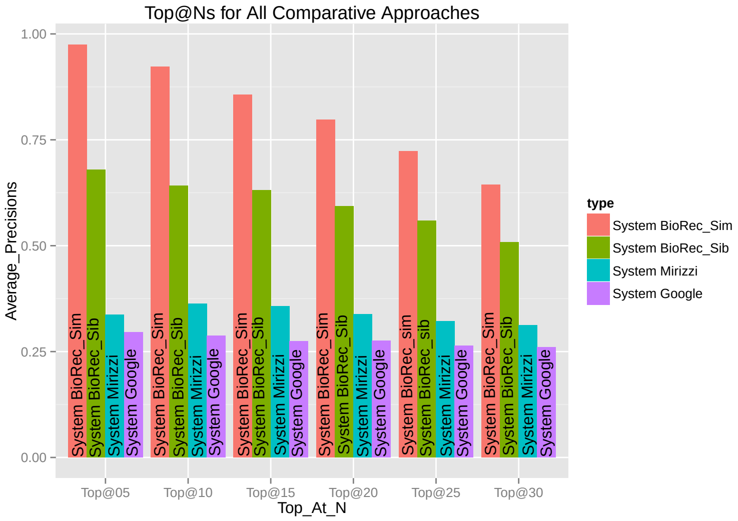


Figure 6.13: Average Precision in All Top@Ns for semantic similar method.

Another comparison was performed using this metric, to assess the success of our approach in **task 5** (appendix F.2), which was asking the participant to select a specific interest from the user profile on which the participant wants recommendations. As shown in figure 6.14, BioRec_Sim outperformed most other approaches by a wide margin in all thresholds in this task. This task was tailored to examine this property in BioRec_Sim and BioRec_Sib, and it showed that BioRec_Sib is not quite as accurate as BioRec_Sim. But, BioRec_Sib did well compared to Mirizzi and Google. Mirizzi did not employ the sibling or semantic similarity relations and Google used no semantic relation at all. Then, the Mirizzi system did not support the specialist with good results where it had a dramatic difference compared with BioRec_Sim and BioRec_Sib. It had very close results to Google in all thresholds, except at Top@30 where Google was better than Mirizzi. This can show the weakness in the way that is used in Mirizzi's system to exploit semantics between different resources. Moreover, it shows the importance of considering an ontology to represent the user profile, which can support the user with more accurate recommendations in such point. Also, it shows the benefit of the method that used to make the recommendations exclusive on a specific interest in the user profile, which is not supported in both (Mirizzi and Google) systems. Thus, both (Mirizzi and Google) systems provide recommendations based on all preferences stored in the user profile, which makes recommendations less accurate. Finally, Google was the weakest in this comparison, showing the importance of using semantic relations such as sibling and semantic similarity to enhance the accuracy of the provided recommendations. However, Google was not the worst in Top@30, this can be explained by the incorrect way of exploiting semantic relations that was used in the Mirizzi system.

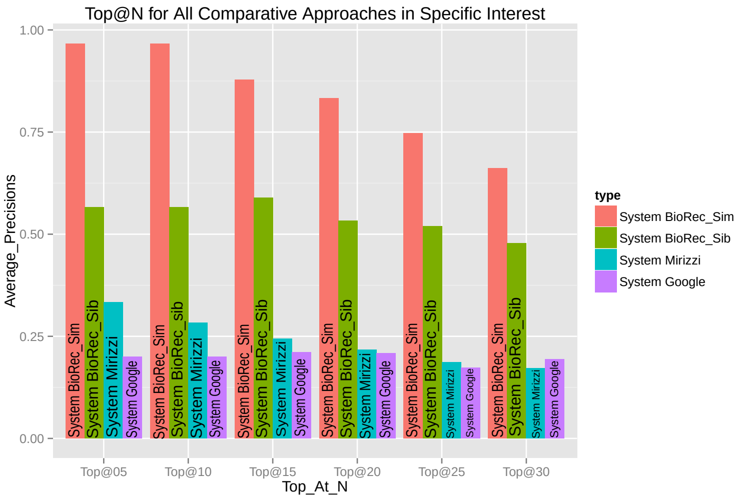


Figure 6.14: Top@N for All Comparative Approaches in Specific Interest.

Moreover, the t-test [219] was applied and BioRec_Sim, achieved significant results in some thresholds. For example, at Top@5 with BioRec_Sib 0.0000655 which is less than 0.01 and represents a significant result; and with Mirizzi Top@5 with score 0.000023 and Top@15 was 0.0000527, where these scores are significant since they are less than 0.01. Also, with Google it had registered at Top@5 0.000005461, Top@10 0.000000013 which less than 0.01 and they are significant.

6.4.6.2 Mean Average Precision Metric

The MAP metric was used to confirm that all results reached by Precision@N more accurate and correct. Thus, the results of this experiment, given in figure 6.15, show a comparison of the four systems. System BioRec_Sim outperformed the other systems, reflecting the fact that the use of the semantic similarity method enhanced the accuracy of the provided recommendations more than using the sibling relation. Moreover, as shown in figure 6.15, systems BioRec_Sim and BioRec_Sib dramatically enhanced recommendations compared to system Mirizzi. This reflected the fact that system Mirizzi did not have an ontological user profile and did not employ the extracted semantics to enhance the accuracy and provide better recommendations. Finally, system Google registered the lowest result because it provided standard recommendations without considering the ontological user profile or semantic relations and associations.

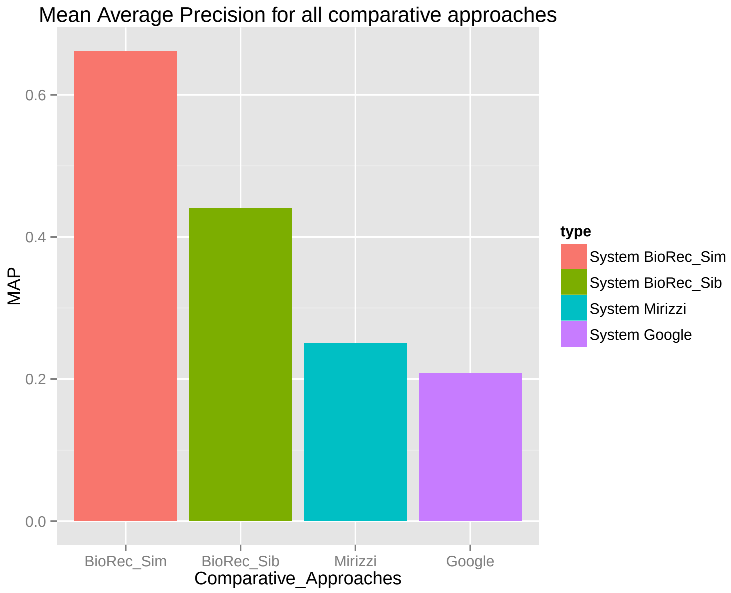


Figure 6.15: Result for applying semantic similar method.

Moreover, a t-test was applied to this evaluation, which registered significant results, including 0.00000051 between BioRec_Sim and BioRec_Sib. This score represents a significant difference on the t-test scale because the value is less than 0.01. Also, the score between BioRec_Sim and Mirizzi was 0.00000012, and the score between BioRec_Sim and Google was 0.00000000205. This reflects a significant result because the value is less than 0.01.

Table 6.5: Mean Average Precision for Each Assigned Task

Systems	Task 1	Task 2	Task 3	Task 4	Task 5	Task 6	Task 7	Task 8
BioRec_Sim	0.7160	0.6739	0.7147	0.6084	0.6988	0.6395	0.5967	0.6466
BioRec_Sib	**0.4162**	0.5502	**0.4294**	**0.4448**	**0.3753**	**0.3961**	0.4740	**0.4426**
Mirizzi	**0.2317**	**0.3125**	**0.2961**	**0.2788**	**0.3172**	**0.2751**	**0.2347**	**0.0567**
Google	**0.2381**	**0.2827**	**0.1943**	**0.2382**	**0.2860**	**0.1777**	**0.1764**	**0.0756**

Furthermore, another test was applied which is more restrictive than the t-test and determines the significant result called Mann-Whitney U-Test [219]. This metric achieved similar results as the t-test, ensuring that all t-test results were accurate and correct. Thus, all registered scores were significant between BioRec_Sim and BioRec_Sib; and BioRec_Sim and Mirizzi; and BioRec_Sim and Google where all of these approaches registered Z score equals 3.3082 and p-value 0.00047 and this is considered as significant when $p \leq 0.01$; and U score equals 0 and the critical value of U equals 9 at $p \leq 0.01$ where this means it is significant[1]. Table[2] 6.6 presents the significant results in each assigned task in more detail.

Table 6.6: Mean Average Precision for Each Assigned Task for Mann-Whitney U-Test significant Results

Systems	Task 1	Task 2	Task 3	Task 4	Task 5	Task 6	Task 7	Task 8
BioRec_Sim	0.7160	0.6739	0.7147	0.6084	0.6988	0.6395	0.5967	0.6466
BioRec_Sib	**0.4162**	0.5502	**0.4294**	0.4448	**0.3753**	0.3961	0.4740	0.4426
Mirizzi	**0.2317**	**0.3125**	**0.2961**	**0.2788**	**0.3172**	0.2751	**0.2347**	**0.0567**
Google	**0.2381**	**0.2827**	**0.1943**	**0.2382**	0.2860	**0.1777**	**0.1764**	**0.0756**

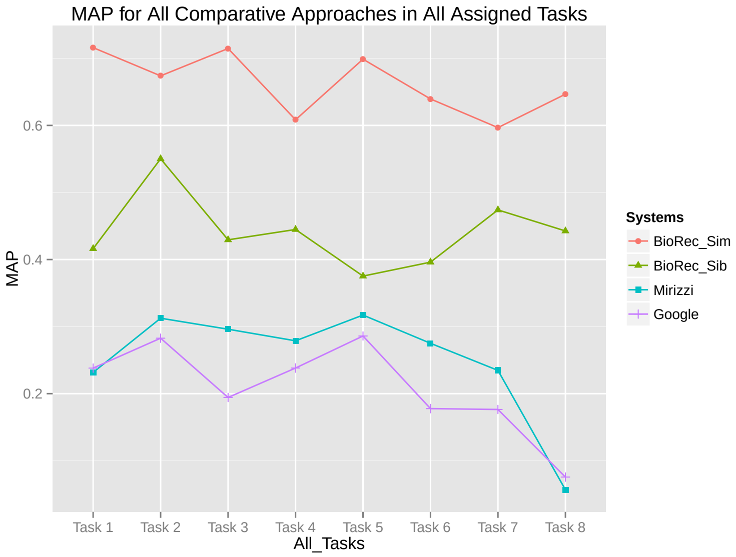


Figure 6.16: MAP for All Tasks in Each Approach for Semantic Similarity Method.

Additionally, figure 6.16 demonstrates the performance of all comparative approaches in all assigned tasks. As shown, system BioRec_Sim outperformed the other comparative approaches in all assigned tasks. However, we will discuss the

[1] `http://www.socscistatistics.com/tests/mannwhitney/`

[2] All records in bold in all tables are representing significant scores.

increase or the decrease of comparative approaches' performances that happened in each task. **Task 1** appendix F.2, shows that BioRec_Sim achieved the highest MAP score. This supports our hypothesis that exploiting semantics such as semantic similarity relation with ontological user profile can help the recommender system make accurate recommendations about general topics in bioinformatics. Moreover, the sibling relation as exploited by BioRec_Sib did well in this task. However, it was still weak compared with the semantic similarity relation to enhance the quality of the provided recommendations. The Mirizzi system was the worst one in this task, this supports our criticism of this approach, that it does not use semantic relations efficiently. They just extracted them without performing further inference to discover new relations or hidden associations that could support the recommender system with extra information to enhance its utilisation. Also, system Google did not have good performance in this task, since it did not use semantics.

Then, in **task 2** we can observe that all approaches had an enhancement in their performance except BioRec_Sim, which had a decrease in its performance, however, it is still better than the other approaches. The reason behind this decrease may relate to the weakness of semantic similarity relation in some types of recommendations such as the recommendations in this task. The enhancement in the other approaches especially the sharp increase in BioRec_Sib approach, where it shows the strength of sibling relation in these types of recommendations. The Mirizzi system is stronger here, but still unable to compete with BioRec_Sim or BioRec_Sib and this is because they use regular user profiles (not ontological) and did not exploit semantics between different resources successfully. Finally, the Google system did better, but still not as well, because it did not consider semantics to support it to provide better recommendations.

In **task 3**, the participant was expecting recommendations that provided him articles which discuss programming languages that he used to use or read about them in the domain of bioinformatics. So, this task is a bit tricky since it requires semantics to support the recommendation process with information that could provide more accurate recommendations. Therefore, system BioRec_Sim did well in this task and it had an increase in its performance compared with task 2. Since, it considered both semantic relation (i.e. semantic similarity) and the ontological user profile, where these two methods help this approach outperform the other comparative approaches. All the other approaches had witnessed a decrease in their utilisation, even though two of them consider semantics and user profiles. For instance, BioRec_Sib was considering sibling and ontological user profile, but sibling is not as good as the semantic similarity relation. Also, Mirizzi was applying semantics and user profile, but the user profile was not ontological and the semantics were not fully utilised. Google was the worst because it lacks semantic support in this type of recommendations.

Furthermore, **task 4** saw a decrease in BioRec_Sim and Mirizzi and an increase in the other approaches, for two reasons: i) for the BioRec_Sim system, the semantic similarity method was not efficient in such type of recommendations and it caused this decrease in its performance compared with the previous task; ii) the Mirizzi system, suffered from the weakness of the user profile, so it usually caused it to provide inaccurate recommendations. Systems BioRec_Sib and Google had an enhancement in their performance. The former was supported by the sibling relation, which sometimes is doing well in this type of recommendations. Since, it supports specialists with articles that discuss gene or protein from the same family that he used to read about. For the latter (Google), sometimes standard recommendations had some enhancement in its performance, but this performance was not consistent in all performed tasks.

Task 5 witnessed an increase in the BioRec_Sim MAP score. This shows the improvement that could be gained when applying the semantic similarity relation as well as the ontological user profile, to recommend to the bioinformatician articles that related to specific interests that he had selected. However, BioRec_Sib was not as accurate when it was exclusive on a specific interest. Mirizzi and Google increased their performance, however, they still could not provide accurate recommendations such as those provided by the BioRec_Sim or even BioRec_Sib systems. This is because both systems (Mirizzi and Google) are not equipped with the method that allows the user to determine a particular interest on which to receive recommendations.

Task 6 showed a decrease in all approaches' performance in comparison with the previous task except the BioRec_Sib, which increased because the extent of preferences became wider and not only in the specific interest of this approach. The weakness in other approaches' performance comes as a result of the bioinformatician's query absence in this task, which supports the recommendation process in the other tasks. BioRec_Sim still performs best here, confirming our main hypothesis in this thesis that exploiting this semantic relations (semantic similarity) contributes to enhancing the precision of the provided recommendations.

Tasks 7 and 8 supplement each other, since the same query was used for both; however, they differ from the judgement perspectives. So, in **task 7** the specialist expected recommendations on something related to his preferred gene or protein. But task 8 will evaluate the provided recommendations whether they stem/associate from/with the preferences or specialist's query. This will be discussed in more detail later in this section. In task 7 all the systems have performance decrease except BioRec_Sib which provided results that related to user preferences. But with this decrease BioRec_Sim outperformed it, showing that the semantic similarity relation with ontological user profile enhances the provided recommendations. The other comparative systems were not accurate in such complex task since their used techniques are not good enough for these recommendations.

Task 8 demonstrated a dramatic drop for Mirizzi and Google, because the first approach did not provide recommendations based on inferred relations. It just used the extracted ones, and the second approach did not exploit semantics at all. On the other hand, BioRec_Sim outperformed all other approaches and it registered an enhancement in the performance compared with the former task. Since, it returned accurate results that associated with the specialist's query and preferences. BioRec_Sib showed decrease in the performance compared with the previous task and it was not efficient in providing recommendations that stem from bioinformatician's query or preference. Thus, we can conclude that exploiting the semantic similarity relation is better than the sibling relation in complex tasks.

Furthermore, from this part and downward we will discuss each task in more detail. As described earlier, we assigned users to each system, and each user had to undertake eight well-defined tasks appendix F.2 in order to interact with the system to assess the features feeding into recommendations provided by our approach. These tasks were designed to test recommendations, starting with performing general tasks and moving gradually to more specific tasks. Now we will discuss the performance of each system in each assigned task in more detail. Table[1] 6.5 provides the Mean Average Precision (MAP) results for each system in each assigned task.

The **task 1** (appendix F.2) was designed to check the level of accuracy on recommendations that can be provided about general ideas about bioinformatics such as definitions, history etc., which can help bioinformaticians to refresh their knowledge about different concepts in this field. As shown in figure 6.17, system BioRec_Sim outperformed the other systems. This shows the important role played by the semantic similarity method and reflects its ability to provide more accurate results when considering this relation in our recommender approach. Surprisingly, system BioRec_Sib had a dramatic decrease compared with BioRec_Sim, where such a decrease can show that the sibling relation is not sufficient as the semantic similarity relation in providing recommendations on general topics. However, it is still sufficient in comparison with the remaining systems (i.e. Mirizzi and Google). System Mirizzi was the weakest system in this comparison, which reflects that such a system is not correctly exploiting semantics between different resources. Moreover, the user profile used, was not constructed based on an ontology, so these reasons can lead to inaccurate recommendations that could be provided by this system. System Google was the second weakest system, which shows that our baseline is good enough to recommend users with general topics about bioinformatics. However, it still had insufficient performance compared to systems BioRec_Sim and BioRec_Sib.

[1] All records in bold in this table are representing significant scores.

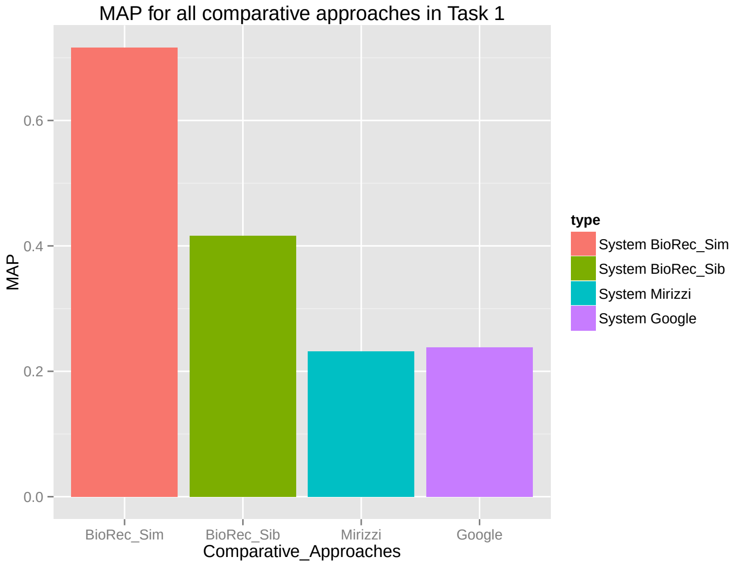


Figure 6.17: Mean Average Precision for All Comparative Approaches in Task 1.

Task 2 (appendix F.2), which was designed to be more specific than the previous one. It tried to assess the performance of the comparative approaches in providing recommendations that support the user with articles that discuss some tools used in bioinformatics such as annotation, sequencing etc. Figure 6.18 demonstrates that system BioRec_Sim satisfied the highest score in this comparison. This can lead to concluding that applying the semantic similarity method with an ontological user profile and search contributes to enhancing the accuracy of the provided recommendations. Then, system BioRec_Sib, which was the second best system in providing recommendations on articles that discussed tools used in bioinformatics, shows that exploiting sibling relations with the ontological user profile performs well for such recommendations. However, it was not as effective with these types of recommendations, such as when using system BioRec_Sim. After that, system Mirizzi, which had a dramatic drop in comparison with system BioRec_Sim and BioRec_Sib. This can lead to two main conclusions: i) system Mirizzi did not successfully employ semantics between resources to discover new information and relations; and ii) the user profile used here was not ontological, which results in inaccurate recommendations because ontologies can enrich user preferences with extra information that helps enhance the quality of the provided recommendations. Finally, system Google, which had a slight decrease in comparison with system Mirizzi, was a result of providing standard recommendations that were less accurate than those systems that consider semantics for providing recommendations.

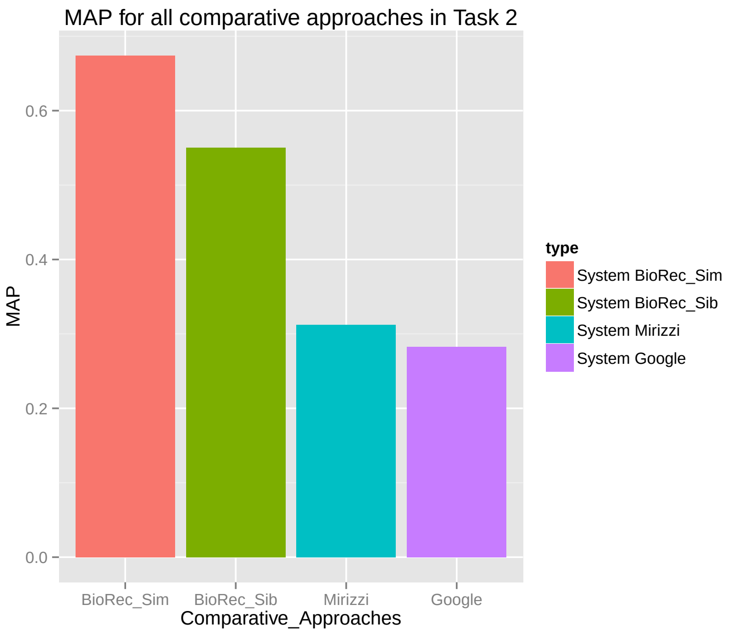


Figure 6.18: Mean Average Precision for All Comparative Approaches in Task 2.

Task 3 (appendix F.2) was tailored to examine the utilisation of the comparative approaches in providing recommendations on articles that discussed or mentioned some programming languages that are popular in bioinformatics. Figure 6.19 illustrates the performance of each system in this comparison, where system BioRec_Sim outperformed the other systems. This reflected the level of enhancement in the provided recommendations. This could be achieved when considering both the ontological user profile as well as the semantic similarity method into our recommender approach. Moreover, system BioRec_Sib had a dramatic decrease in comparison with system BioRec_Sim. This indicated that considering the sibling relation and the ontological user profile into our recommender approach is not as effective as the relation considered in the system BioRec_Sim. Then, the systems' performance gradually decreased when looking at the other systems (i.e. Mirizzi and Google). This decrease occurred for two reasons: i) system Mirizzi did not have a solid inference that can reason through our resources to find new programming languages that could be recommended to the users. This system did not have an ontological user profile that inferred new information based on the information overlapping between user preferences and ontology concepts; and ii) system Google, which provided typical recommendations that were inaccurate, did not exploit semantic relations to enhance recommendations.

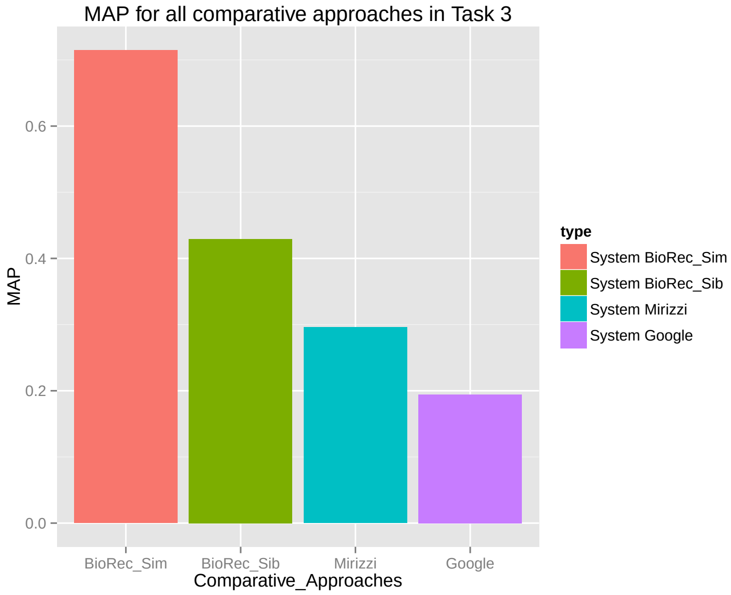


Figure 6.19: Mean Average Precision for All Comparative Approaches in Task 3.

Task 4 (appendix F.2) was designed to be more sophisticated than the previous tasks because comparative approaches were examined to assess their accuracy in providing the user with recommendations about specific ontologies that he was used to reading articles about, such as GO, PO etc. As shown in figure 6.20, system BioRec_Sim achieved the highest score in comparison with the other approaches. This illustrated the level of success that can be achieved when considering semantic similarity relations and the ontological user profile to recommend the user with articles that mentioned or discussed ontologies that are more relevant to the ontologies that he was used to reading about. Then, system BioRec_Sib, which had an average decrease in comparison with BioRec_Sim, also reflected that considering the sibling relation as well as the ontological user profile performed well in such types of recommendations. Surprisingly, system Mirizzi had a dramatic decrease that occurred as a consequence of the inaccuracy of this approach in exploiting semantics between our different resources. Also, it did not consider an ontology in formulating the user profile, which enriched the profile with extra information gained from the overlapped information between the user's preferences and ontology's concepts. Finally, system Google was the weakest approach in this comparison, which shows the importance of considering semantic relations to enhance recommendations, which were absent in this approach. Thus, the former reason caused this weakness in system Google.

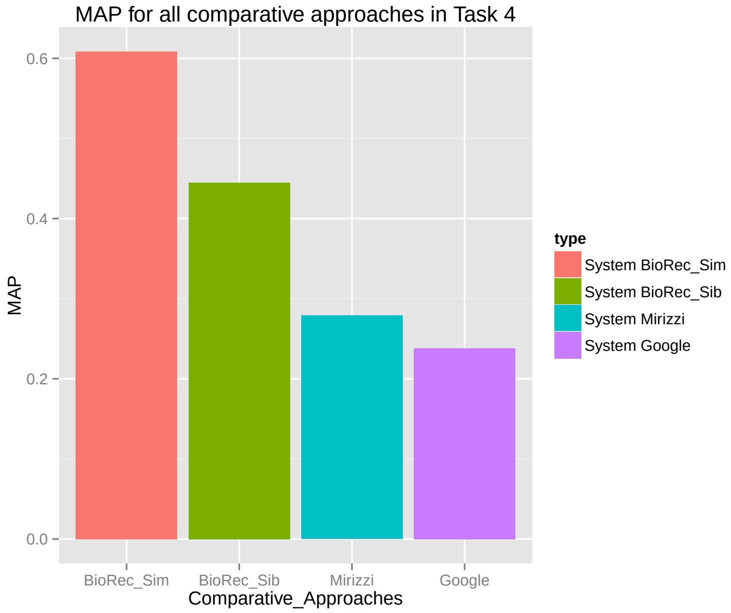


Figure 6.20: Mean Average Precision for All Comparative Approaches in Task 4.

Another example is that of **task 5** (appendix F.2), which was designed to ask the user to assess the provided recommendations when he selected a specific preference to have recommendations for and specifically on that interest only. So, as shown in table 6.5 and figure 6.21, system BioRec_Sim exhibited the highest MAP score. This reflects the fact that applying the semantic similarity method in our approach enabled more suitable recommendations to the user on the most relevant content to be provided. The other three comparative systems had a significant difference in MAP in comparison to BioRec_Sim. This result may reflect that the sibling relation is not sufficiently effective on its own to provide recommendations on specific interests. But, it still has a significant difference compared with the other systems (Mirizzi and Google), which did not include considering sibling relations. Mirizzi and Google systems are also not supported with the method that allows a user to select a specific interest; even though the user can select a specific interest (because all comparative approaches have the same interface) on which to receive recommendations, the provided recommendations are based on all his preferences. This can confirm the advantage of applying such a method to our recommender system, which exploits semantic relations (sibling or semantic similarity).

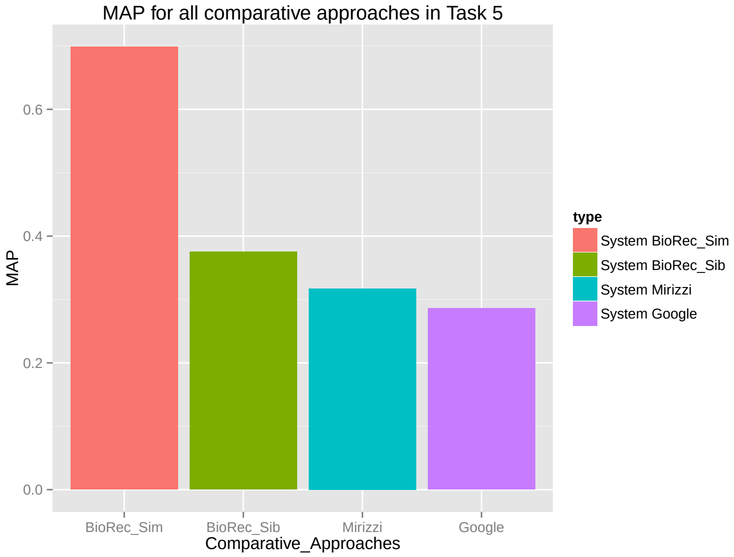


Figure 6.21: Mean Average Precision for All Comparative Approaches in Task 5.

Task 6 (appendix F.2) was designed to assess the accuracy of the provided recommendations without considering the user's query. Thus, as shown in figure

6.22 and table 6.5, system BioRec_Sim outperformed the other systems. This result reflects the effectiveness of applying both semantic similarity methods and the ontological user profile into our recommender approach, which contributes to enhancing the accuracy of the provided recommendations. In the same task, system BioRec_Sib demonstrated a significant decrease in performance in comparison to system BioRec_Sim, which demonstrates the important role that the semantic similarity method plays in enhancing recommendations. System Mirizzi has a significant difference compared with BioRec_Sim, and this reflects the effectiveness of considering semantic relations (such as the semantic similarity method) as well as an ontological user profile. Although the difference with system BioRec_Sib is not as great as with BioRec_Sim, this still demonstrates that exploiting sibling relations also contributes to enhancing the recommendations. However, it does not provide as significant improvement as that provided by the other semantic method. Moreover, the results suggest that an ontological user profile can enrich the user's preferences with valuable information that can help to provide recommendations for the most relevant content. The enrichment of the user preferences will involve calculating the cosine similarity between his preferences with the ODP concepts. So, sometimes the ODP concept has extra information or relation that helps the preference connect to the best match concept from the inferred semantic network. Since the semantic similarity relation will be used to enhance recommendations based on this connection. Thus, whenever user preference is very rich with information and relations acquired from the ODP concept, the task of finding the best match concept from the inferred semantic network will be much easier and more effective. System Google, which was the weakest system in this comparison since it does not exploit semantic relations, does not consider the ontological user profile, and it provides standard recommendations.

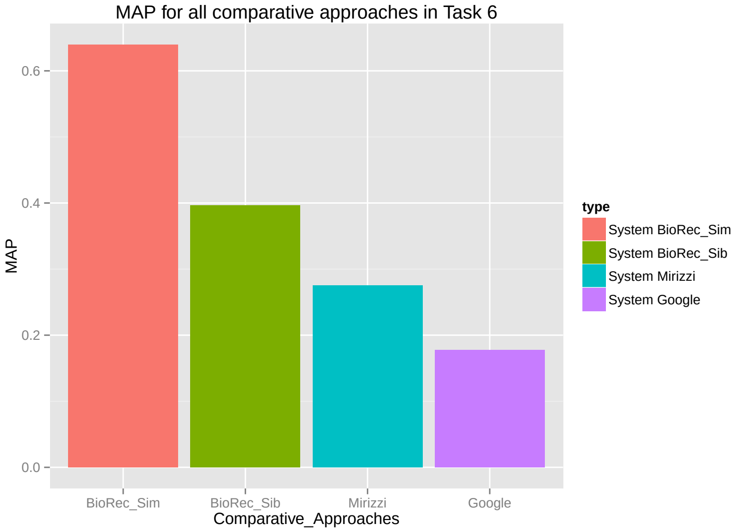


Figure 6.22: Mean Average Precision for All Comparative Approaches in Task 6.

Finally, **tasks 7 and 8** (appendix F.2) were designed to complement each other. For instance, **task 7** was designed to recommend articles about a specific gene or protein that the user was used to reading about, and he was expected to receive recommendations on articles that discussed something relevant to the same gene or protein. As shown in figure 6.23, system BioRec_Sim outperformed the other approaches, and then system BioRec_Sib achieved the second highest MAP score in this comparison. The success for these two systems was achieved as a result of considering both semantic relations (i.e. semantic similarity and sibling) and ontological user profiles that contributed to enhancing the accuracy of the provided recommendations. However, system BioRec_Sim had better performance than BioRec_Sib, which demonstrates that the semantic similarity relation can provide more accurate recommendations than those provided by the sibling relation. Then, system Mirizzi had a dramatic decrease in comparison with BioRec_Sim and BioRec_Sib. This shows the importance of both exploiting semantic relations between different resources and ontological user profiles in providing recommendations; where these properties were not considered or partially considered in Mirizzi's system. They help to provide more accurate recommendations. Then, system Google was the weakest system in this comparison because it provided standard recommendations and did not take into account the semantic relations and associations between different resources.

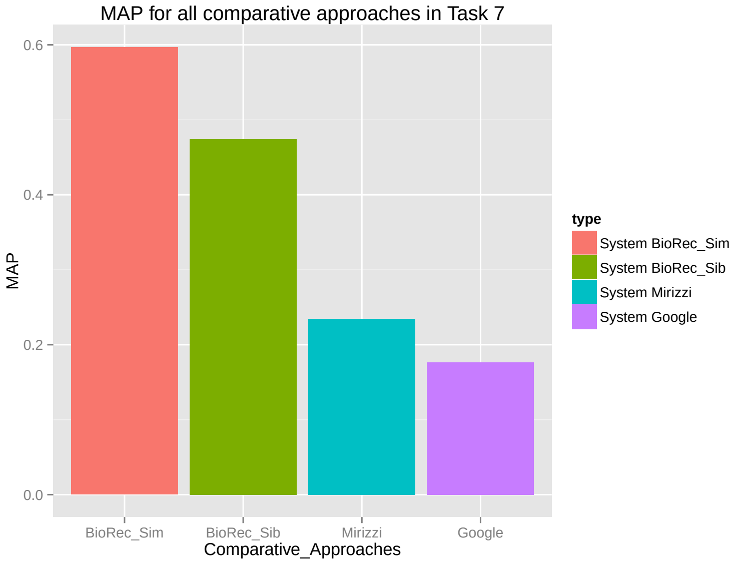


Figure 6.23: Mean Average Precision for All Comparative Approaches in Task 7.

Although **task 8** used the same query that was submitted in task 7, it was designed to assess results based on only two features: i) whether the shown recommendations associated in some way with the user's submitted query and preferences (to assess the semantic similarity relation); or ii) whether the shown recommendations stemmed from the user's submitted query and preferences (to assess the sibling relation). For instance, if the specialist used to read about a specific protein, then he submitted the name of that protein to get recommendations associated with it, which means the recommendations should satisfy high similarity score for the submitted query and his preferences. This will ensure that all recommendations are semantically similar with the specialist's protein and we have asked the participant in this task to assess the provided recommendations based on this idea. Furthermore, to assess the sibling relation we have asked the specialist to submit a gene or protein name he used to read about. Then, the participant will assess the provided recommendations, and whether they have articles that discuss protein or gene stemming from the same family of the submitted gene or protein. In this case, it is not compulsory that the recommended gene or protein has similarity with the submitted query or user profile preferences.

As shown in figure 6.24, system BioRec_Sim outperformed the other systems. This reflected the high quality of the provided recommendations, where this approach was able to provide the user with associated articles, or in other words, articles that are semantically similar to both user query and preferences. Moreover, system BioRec_Sib showed an average decrease in comparison with BioRec_Sim. This strengthened our hypothesis that exploiting semantic similarity relation can enhance the accuracy of the provided recommendations more than exploiting the sibling relation. In contrast, system Mirizzi had a surprising dramatic decrease that shows the weakness of this approach in providing recommendations in such relations (i.e. sibling and semantic similarity) because it did not exploit triples to infer for semantic relations and information. For this reason, it was the weakest system in this comparison, even with system Google, which provides standard recommendations. Finally, system Google also showed a dramatic decrease in comparison with BioRec_Sim and BioRec_Sib because it did not consider semantics when providing recommendations. However, it was still better when compared with system Mirizzi in providing recommendations for articles that stemmed from or were associated in some way with the user's query and preferences.

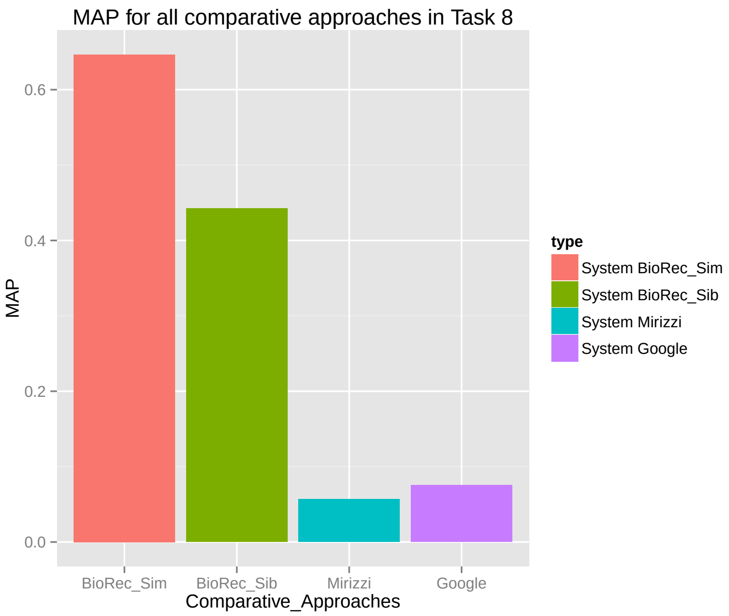


Figure 6.24: Mean Average Precision for All Comparative Approaches in Task 8.

6.4.6.3 Mean Absolute Error Metric

The MAE metric was also considered to assess the level of predictive accuracy in all comparative approaches. As shown in figure 6.25, system BioRec_Sim outperformed the other approaches where this can demonstrate the effectiveness of employing both semantic enrichment with semantic similarity relation and ontological user profile in enhancing the predictive accuracy. Then, the level of error gradually increased until reaching the Google system, which is our baseline. For system BioRec_Sib, the increase in the error level represented evidence that exploiting semantic similarity relation can provide more accurate recommendations that have better predictive power than using sibling relation. Further, both BioRec_Sim and BioRec_Sib performed better than Mirizzi. Because of the weakness of this approach in exploiting semantics between our resources as well as the effectiveness of the ontological user profile that is used in both approaches (BioRec_Sim and BioRec_Sib), is better than regular user profile that is used in the Mirizzi system. However, Mirizzi still did well compared with Google, which was not employing semantics at all in the provided recommendations, making Google the weakest of the systems.

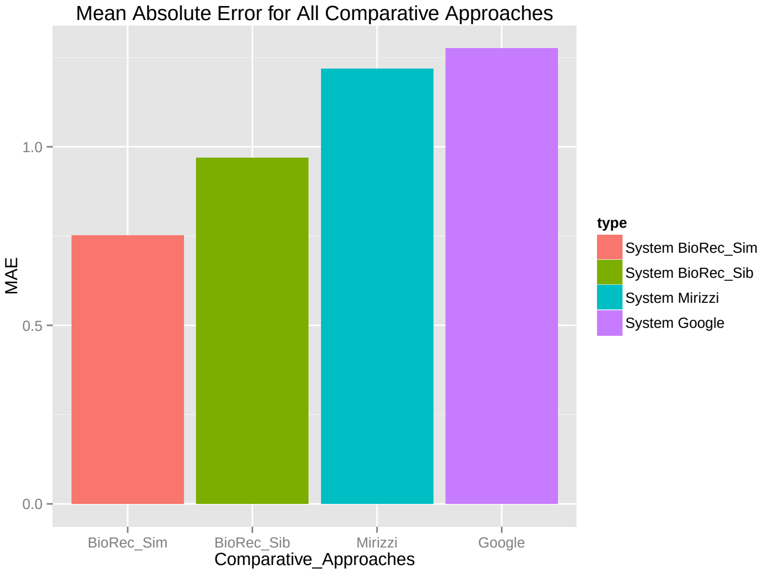


Figure 6.25: Mean Absolute Error for All Comparative Approaches.

6.4.7 Experiment Summary

Finally, after all comparisons in the previous section and based on the results and the data analysis, we conclude that our approach is novel in the type of exploited relation, semantic similarity to support specialists in specific domains such as bioinformatics and enhance the accuracy of the provided recommendations. Also, it supports specialists by using these semantic relations while employing an ontological user profile to tailor recommendations to each user individually based on his preferences. So, it demonstrates its importance by enhancing the quality of the provided recommendations based on the reached results. Moreover, it compares well with Mirizzi et al. [159], which used semantics and user profile to provide recommendation. But, without further discovering and inferencing through the extracted relations and information to find more hidden relations or information such as our founded relation which can be exploited to enhance the recommendations. In addition, Mirizzi et al. [159] lacked the ontological user profile which can add extra information to the user profile and help to find most relevant content. Moreover, it compared with our previous method, exploiting sibling relation that supported with the ontological user profile, and showed that it is able to better enhance the accuracy of the provided recommendations than what the sibling relation has done. Also, it compared well with Google, our baseline and it showed that exploiting such relation (semantic similarity) can add more enhancement than a general method. This distinguishes our method from all others.

6.5 Conclusions

In conclusion, this chapter discusses the evaluation methods that have been applied over all our two experiments. These experiments were designed to examine our contributions of this thesis, whether they have added an improvement for recommender approaches in specific domains. Also, they helped specialists, using their search activities and preferences to provide them with more accurate recommendations, to help them discover new information that could not be achieved without using our recommender approach which had supported by semantic-based techniques that enhance its performance. Thus, i) the first experiment assessed our contribution which considered the sibling-inferred semantic relation and applied it to a content-based recommender system that was supported with an adaptive ontological user profile to enhance the precision of the provided recommendations. So, based on the results and data, we can say that our method is novel in exploiting the sibling semantic relation as well as an adaptive ontological user profile to enhance the accuracy of the provided recommendations in a specific domain such as bioinformatics. Then ii) the second experiment was designed to assess our contribution when applying our semantic similarity relation, acquired through the inference process which has involved multiple resources. Our hypothesis in this experiment was exploiting such a relation as well as the ontological user profile in the content-based method helps enhance the accuracy of recommendations. It outperformed the level of improvement that could be reached when considering the sibling semantic relation and the methods which applied by the other approach discussed in the literature to enhance the accuracy of the provided recommendations. Thus, based on this experiment and its results, we can say that our recommender method is the first method that applied such relation which worked side by side with an ontological user profile to enhance the precision of recommendations tailored for specialists in bioinformatics.

Chapter 7

A Critical Analysis

7.1 Introduction

This chapter provides a critical analysis for the work that has been discussed in this thesis. It concludes with several strong points that can be used to distinguish this work from other relevant works. Moreover, it addresses all the weak points from which our work is suffering. It also shows the implication of the gained results from the proposed methods. Thus, the comprehensive view and critical analysis in this chapter will help draw a final conclusion about all the semantic-based methods that have been designed and discussed in this thesis.

7.2 Results and Findings Analysis

Based on the results of the two experiments that were discussed in the previous chapter. In addition, based on all the methods that we have suggested or developed thus far, the efficacy of our proposed semantic techniques and how they enhance the recommendations, becomes clearer when they are exploited in recommender systems. For instance, in terms of the inferred semantic relations (sibling and semantic similarity) that were gained as a result of the overlapping between multiple bioinformatics resources, the sibling semantic relation was able to provide an enhancement of the recommendations by recommending the user with articles that discuss his/her submitted query, preferences and articles that contain concepts which have sibling relation with the preferences stored in his/her profile. The main weakness in this relation involves selecting the sibling concept when we exploit the sibling relation. Our method was designed to exploit the first inferred sibling for a specific preference in the user profile, whereas sometimes others like the second, third, fourth, etc. sibling may be more relevant to the particular preference in the user profile. This may cause some weakness in the accuracy of the recommendations and make exploiting sibling relation inaccurate with some assigned tasks in both prior experiments. Moreover, this is also one of the reasons why exploiting semantic similarity promises better results than exploiting sibling relations. To overcome this problem, a method should be employed after inferring all siblings for all concepts and representing the inferred relations to organise the siblings of each concept in the semantic network based on their relevancy to the specific concept. This will help our method exploit the most appropriate or relevant sibling to provide recommendations on those articles located in the top (the first) sibling for the specific preference in the user profile.

Furthermore, in the second experiment 6.4, which compared exploiting semantic similarity, sibling, an approach from the literature [159] and Google API particularly in task 5, which required the participant to select a specific preference in order to have recommendations on it. We can observe the dramatic decrease that occurred with the approach of exploiting sibling relation, which is called BioRec_Sib. This decrease can be interpreted as happening because the first sibling of the selected preference for most participants is not the most appropriate sibling. This also emphasises the problem mentioned previously, where our framework should have a method that is in charge of organising the siblings for every concept based on their relevancy to it.

There is another issue that we should be aware of regarding the previous experiments. Although the results show that our approach in both exploited semantic relations was highly effective and enhanced the accuracy of the provided recommendations, these experiments were conducted over small groups of participants. As mentioned in [189], it is difficult and quite expensive to find subjective participants in the field of recommendation to participate in a user-centric evaluation such as ours. Thus, these results may become weaker or less accurate when the number of involved participants is increased.

There is another point related to the sibling relation exploitation in experiment 6.3. In spite of the enhancement that can be gained from exploiting sibling relation in our framework as has been shown in the results of our approach, which is called BioRec_Full. This approach employed semantic relation (i.e. sibling) as well as an automatic adaptive ontological user profile, and it outperformed all other comparative approaches in terms of classification and prediction. However, when we look at the approach that exploits the user profile only (i.e., BioRec_SN) and the approach that exploits the semantic network or sibling semantic relation only (i.e. BioRec_SN), we can conclude that exploiting the user profile outperformed exploiting the semantic relation in most assigned tasks. This confirms that exploiting the sibling only without the user profile is not expected to satisfy a high level of enhancement to the provided recommendations. It shows the important role that can be played by the user profile to enhance recommendations.

In the second experiment, 6.4, in terms of constructing the user profile, users' preferences were collected explicitly. This caused some weakness in the accuracy of the provided recommendations because the explicit collection of user preferences is sometimes not quite as accurate as implicit collection. Since, the latter helps us draw an accurate conclusion about which user prefers to read what by observing their behaviour for a long time (i.e., one month or more). However, the former is dependent on the user's accuracy when he/she is filling his/her preferences; because some people are not quite as accurate in this or because they sometimes forget some of their preferences. This may cause weaknesses in the provided recommendations. We had to consider this because of the limitation regarding time and to perform the experiment as quickly as possible. Thus, if we had used the other way to collect user preferences implicitly like the first experiment 6.3, then our approach (which exploited sibling or semantic similarity) would have definitely satisfied better results and provided more accurate recommendations.

Furthermore, based on the analysis of the two experiments (6.3 and 6.4) in the previous chapter that we conducted to measure the accuracy of our approach in terms of classification accuracy, we can conclude that our recommender approach in both exploited relations (sibling and semantic similarity) returns good results and enhances recommendations for the short term (such as Top@5,10,15,20,25,30). But our baseline (Google) is better for long-term results (such as top@60, 65, 70, etc.) and weaker in the short term. Thus, such problems represent a weakness in our recommender approach. We can conclude from this that for general searches and recommendations, Google returns better results, but for specialist searches and recommendations, our proposed semantic methods will return better and more accurate results for both searches and recommendations.

Finally, this research as any other research has some limitations and needs some developments as future work. These will be discussed in the following sections.

7.2.1 Research Limitations

As any research, our approach has some limitations, and these limitations are classified into two types: i) limitations that need time to be performed. For instance, converting the processed file from text and XML into OWL files to be processed in our framework and reasoned by the Jena built-in reasoner was not fully automated. Because it requires developer intervention, and this may cause some errors during the converting process as a result of this intervention. Moreover, the problem of updating some resources that are not in OWL and need to be converted to OWL. Since, they were used in formulating process of our inferred semantic network, where the current update method is only exclusive in the resources represented in OWL format such as GO and PO. Therefore, to overcome all aforementioned problems, a method should be developed that is in charge of converting text and XML files into OWL files in order to be processed by our approach. In addition, there are further potential tests and evaluations that could be performed on some of the discovered and extracted relations from our different resources that were not considered because of limitations on time. Specifically, *Same_As, equivalent, Is_Type_Of, grandSubClassOf and SuperClassOf* where each one of these relations can be assessed by a separate experiment and compared with other relations (sibling and semantic similarity) and with the works from the literature. To ensure that it can help in enhancing the accuracy of the provided recommendations.

Another source of weakness in this study is associated with further analysis that should be considered over our different resources. This may lead to discovering a new relation. The discovered relation may make us infer new facts or information that could help further enhance the accuracy of the provided recommendations.

Then, we can assess the performance of the discovered relation and compare it with both sibling and semantic similarity relations, to see which is better to be exploited and able to enhance the precision of the provided recommendations.

Furthermore, ii) the limitations caused by processed resources and machine performance caused long delays where our machine was not able to read and process all OWL files, and returned with Java heap size problems every time. For this reason, and after several tries, we decided to divide huge files, such as GO and PO OWL files, into several sub-files, even though this meant our machine was not able to read and process all sub-files. For this reason, we considered the maximum number of files from each resource that can be handled by our machine. For more details about the processed files in our datasets, please read section 4.2.2.

Thus, a high-speed machine with bigger RAMs should be considered that is able to overcome and eliminate all machine size and speed problems.

7.2.2 Future Work

There are several developments that could be considered in our approach that will contribute to enhancing the system performance and quality of the provided recommendations. For instance, converting text or XML files into OWL should be done automatically to overcome any difficulties that could be found by the developers or researchers in the reasoning process. Moreover, the user profile can also be a contextual user profile and consider short- and long-term preferences to make recommendations more accurate and help the user show results based on the time and place he/she was in during the recommendation process. For instance, recommendations shown to the researchers in their office should be different from those shown in their labs and houses. Furthermore, adding a new method can determine which semantic relation is more useful for some cases. For example, if using sibling relation will give better results in some points, then this method should shift to sibling relation and provide its recommendations and vice versa. Moreover, discovering and exploiting new semantic relations could be achieved as a result of information overlapping between different resources or as result of analysing the processed resources and measuring the level of relevancy between them. Furthermore, if this work is to progress an alternative platform which is based on big data should be considered to be able to handle the complex data that were processed in this work.

Thus, the existence of all aforementioned developments and enhancements will ensure better utilisation and references for our recommendations that can help researchers enrich their knowledge.

7.3 Comparison between Different Methods

This section will provide a comparison between our different suggested methods and the relevant methods suggested by other works.

(i) **A semantic-based method for specialist search:** This method was designed to reason through different bioinformatics resources, then extract semantic relations and hidden associations to employ them for enhancing the accuracy of the provided recommendations. This method is fully automated, unlike those of [100] and [98], which require developer intervention to complete their reasoning process. Since the former is manually assigning top classes to the reasoner, any mistake will cause several mistakes to occur as a consequence of this assignment. The latter is also not fully automated since it waits for the developer's assertion to decide whether "has part" and "has-part" are equal. In contrast, our method is fully automated and does not have any intervention during the reasoning process.

(ii) **A method for reasoning rules and inference of semantic relations:** This method provided seven semantic rules designed to reason through multiple resources and mine several kinds of data. In this thesis, two of the seven rules were evaluated, called the sibling semantic relation and the semantic similarity relation. The latter will be discusses in the following point. The former, which infers new relations between a couple of concepts even though these concepts are located in different resources. Further, this method is different from [81] and [85] in several aspects, such as the type of the discovered relations and the tool used to discover the relations. Elenius et al. [81] and Rakib et al. [85] employed SWRL for this purpose. This tool has a limitation in handling complex data, which ignores many concepts that may have relations with each other.

(iii) **Method of semantic similarity:** This method was suggested to calculate the semantic similarity between different concepts from multiple resources during the reasoning process and it works side by side with the semantic similarity rule (which provided by the previous method) to satisfy its goal. Moreover, it also considers the concepts' similarities, taking into account the process of computing the semantic similarity. It is unlike the methods that were developed in [81] or [85], which used SWRL rules to discover relations, since these rules are not quite effective with complex relations and data. However, our method used the Jena custom built-in rule that fires during the reasoning processes, leading to the discovery of more similarity cases between concepts and is able to handle complex data. It also differs from the methods of Elenius et al. [81] and Rakib et al. [85] regarding the type of discovered relations. Moreover, this method is also unlike the Teng et al. [62] method, which calculates the functional similarity between GO contexts. Teng et al. [62]'s method did not employ any inference method or semantic rules during the process of calculating similarity. This may contribute to ignoring new similarity cases that may occur as a result of applying an inference method and semantic rules.

(iv) **A method for representing semantic relations and associations was in essence the automatic construction of the inferred semantic network:** This method was designed to represent all inferred semantic relations and associations that have been discovered via the reasoning process. It is also able to beat most of challenges that could be found as a result of inconsistencies between multiple resources. It is different from other methods such as [122] which designed a semantic graph; however, it is still not fully automated since it waits for a developer's or specialist's decision to construct a new level in the created graph. This would be effective for small datasets, but it is not quite handy for large datasets that consist of various concepts and different relations.

(v) **A method for keeping some parts of our inferred semantic resources updated:** This method was designed to keep some of our inferred semantic network resources in the OWL format updated. This will help the bioinformatics researcher read up-to-date information. This method is different from other approaches, such as Sangers et al. [210], who designed a method that can update resources in RDF format only by using the OUL. This language (OUL) suffers from several shortcomings. It is unable to handle namespaces, and it only considers the top concept in the semantic network or ontology to update the ontology. If the top concept does not have an update and the lower concepts do, this method will not be able to capture that update. Our method does all these things. Then, after it downloads the semantic resources that have updates (GO or PO), it sends the reasoner to perform reasoning against the other resources. After that, it just replaces the updated part of the semantic network.

(vi) **A prototype system that provides personalised content-based recommendations:** This prototype system implements of all the aforementioned methods and techniques; it then exploits them to provide a personalised service which is embodied in content-based recommendations. This prototype system is supported with two sub-methods that work side-by-side with the exploited semantic relations intended to enhance the accuracy of the provided recommendations. These methods are i) a method for providing an automatic adaptive ontological user profile for each user to receive recommendations tailored to him/her based on the preferences stored in the user profile, and ii) a method allowing users to determine specific interests to make the provided recommendations are focusing on the selected interest. However, some of the relevant works tried to perform some enhancements on the recommendations by exploiting semantics, such as [159], [161] and [164]. These studies did not successfully exploit semantic relations and information. The first and second works suffered from a couple of shortcomings that did not perform further inference in the LOD dataset (ontology) to discover some new relations and information that may exist between different resources included in the LOD. The third one also suffered from similar problems to the former approaches; it did not exploit some included relations in the LOD dataset, such as has-part, or it did not even discover and exploit relations that may exist as a result of information overlapping.

The aforementioned works claim that they provided recommendations by using the semantics information of multiple resources, but their works lacked exploiting semantics successfully for discovering new relations and information that may help provide better recommendations. Also, when our method was compared with their methods from another perspective, we will find another problem from which these approaches suffer: the weakness of their user profiles. Both (the first and second) approaches provided an automatic user profile only concerned with adding new movies without considering updating and deleting mechanisms to the profile. The user profile was responsible for keeping the user profile up to date and coping with frequent changes that may occur in the user profile during the time. They also did not build an ontological user profile that could contribute to enriching the profile with valuable information that could be exploited to enhance the recommendations. In addition, the third approach suffered from similar problems; however, it was worse than [159] and [161] in terms of the user profile. This approach was not built as an ontological user profile and was not even equipped with an automatic method responsible for adding, deleting and updating, which limits this approach toward providing accurate recommendations. There are other approaches that provide an automatic adaptable user profile for recommendations. But, these approaches cannot compare to ours since they are not concerned with applying semantic relations and information into recommender systems to enhance their performances, which represents the main goal of this work.

7.4 Conclusions

In conclusion, this chapter critically discussed all methods and techniques. Also, it showed all the reasons behind the strength and weakness of our approach in providing recommendations by exploiting semantic relations (such as sibling and semantic similarity) and hidden associations that were gained as a result of the information overlapping between multiple resources.

Moreover, this framework and all the suggested methods included with it are general and flexible enough to be used for any other domain. This framework was tested in the bioinformatics dataset due to the need of this discipline for such applications. However, we have taken into account the generalisation of this framework while we were constructing and designing the different methods included in it. Therefore, this framework could work properly in any other domain, such as math, physics, law, etc.

Chapter 8

Conclusion

8.1 Summary

This work aimed to address some of the challenges and limitations in the field of recommender systems and specialist search, especially in the domain of bioinformatics, since this domain has multiple resources that contain various semantic relations and hidden associations. Moreover, these resources need to be combined in order to extract their contained relations and infer new semantic relations and associations which may exist between them. However, combining these resources and exploiting the discovered and inferred semantic relations and associations included between them is non-trivial. Several challenges and difficulties need to be overcome, such as inconsistent structures and information overlapping between multiple resources. We need to reason through these resources and extract semantic relations and hidden associations and infer any new relation that may appear as a result of information overlapping between these resources. Then, exploit them by supporting specialist search and improving the precision of recommendations on the content (i.e. articles) of bioinformatics. Thus, this work has developed new semantic methods that contribute to addressing most of the aforementioned problems and support the specialists in the field of bioinformatics with accurate recommendations that meet their needs.

This work is novel in the semantic-based methods developed to reason through different bioinformatics resources that contain various semantic relations and associations. Additionally, it was supported with an adaptive ontological user profile, which was employed in specialist search to enhance the accuracy of provided recommendations. Thus, the novelty of this approach can be distinguished by: i) its ability to reason through multiple resources with both structured (such as ontologies) and unstructured resources in the form of corpora to extract semantic relations and hidden associations that may exist as a result of information overlapping between these resources. Moreover, ii) it can also exploit specific types of relations inferred from the overlapped information of different resources, such as sibling relation, where such relations do not exist in the original resources. Then, it can exploit them to work side-by-side with an automatic adaptive ontological user profile and content-based system to enhance the accuracy of the provided recommendations on read content (i.e. articles) in the field of bioinformatics. Furthermore, iii) the new method, semantic similarity, is a new type of relation constructed to be considered by our reasoning method during the reasoning process. In other words, this method will be inferred by calculating the semantic similarity between different concepts while our reasoning performs the inference to find the most semantically similar concept to the processed ones. It works side-by-side with an ontological user profile and content-based to enhance the precision of the provided recommendations in the domain of bioinformatics. It endeavours to be tailored to each specialist/user recommendations based on his/her preferences.

Thus, all former techniques will be used to eliminate Tom's problems (section 1.1 and 1.2), where he will be able to get more accurate recommendations on his preferred content, and these recommendations will be tailored to his preferences. It will also help him discover more information and articles that may help him enrich his knowledge and become aware of all new information added to the field of bioinformatics.

In this chapter, we first motivate and summarise all the developed methods. Then, illustrate the achieved contributions and how these contributions fulfilled the research aim.

8.2 Contributions of this Work

This thesis makes a set of contributions that enhance recommendations through semantic-based techniques in specialist search and multiple resources in the field of bioinformatics. These contributions are as follows:

- **A Semantic-based Method for Specialist Search:** We developed a new method that is able to reason over multiple bioinformatics resources. It then extracts semantic relations and hidden associations to exploit them for enhancing the accuracy of the provided recommendations. This fulfilled the first aim of this thesis, 1.3, was discussed in section 4.2 and was implemented in section 5.4.

- **Reasoning Rules and Inference Semantic Relation:** We developed a set of semantic rules able to extract different types of data that exist among multiple resources. Two of them (i.e. sibling and semantic similarity) are the most promising ones exploited in this work. The sibling relation, which is a novel relation that is able to find a connection between different concepts even if these concepts are from different resources, and the other will be discussed in more detail in the next contribution. The semantic rules satisfied the second aim of this work, 1.3, were presented in section 4.2.1 and 4.2.3 and were applied or used in section 5.4.

- **A Semantic Similarity Method:** A novel method that has been extracted based on information overlapping between multiple bioinformatics resources was developed. This method is the first method to calculate semantic similarity during the inference process, which may lead to discovering new semantic similarity cases that may not appear in the normal way. It was in charge of fulfilling the second aim of this thesis, 1.3, was discussed in section 4.2.3.7 and 4.4 and was implemented in section 5.6.

- **A Method for Representing Semantic Relations and Associations:** The method for representing semantic relations and associations was in essence the automatic construction of the inferred semantic network. Moreover, it maintains our inferred semantic network up-to-date, if any update happened in the semantic resources. This method is novel in the way of addressing the many challenges and inconsistencies that may appear as a result of incompatibility between different resources that have different structures and relations. This fulfilled the third and fourth aims of this research, 1.3, was illustrated in section 4.3, 4.5 and performed in section 5.5 and 5.7.

To this end, we have satisfied all the mentioned contributions, specifically the main contribution which provides a semantic-based method that is able to reason through different resources and extract semantic relations and associations. Then, it exploits sibling and semantic similarity relations in order to enhance the accuracy of the provided recommendations. As a result of this, the main aim and objectives, section 1.3, of this thesis have been fulfilled. This was done by introducing the semantic method that is able to reason through multiple resources and extract different types of relations such as sibling, semantic similarity, sameAs, is_a, equivalent, etc. Then, it exploits two of them (sibling and semantic similarity), which were the most promising relations, to enhance the accuracy of the provided recommendation. The former relation shows how our approach is able to connect two different concepts that exist in multiple resources with each other through a new relation called sibling. The latter relation shows that the relation between concepts can be captured from other perspectives. It is not necessary for these concepts that have a semantic similarity relation to have a direct relation if they are in same resource or even in different resources. Moreover, this relation is supported with a method to calculate semantic similarity during the inference process and by considering concepts' similarities and semantic similarity scores, which help find more semantic similarity cases. After that, a method that is able to defeat most inconsistencies between different resources' structures to represent all inferred relations and information in form of semantic network is presented. Also, it is concerned with contacting the original resources of some part of the inferred data (i.e. OWL) in the semantic network periodically to consider any change or update that may happen in their resources. Finally, user profiles mapped with the most relevant concept in the semantic network to provide each user with enhanced recommendations that are individually based on his/her preferences and on the exploited relation sibling or semantic similarity.

Lastly, with all the provided methods and techniques, this work can be considered the first step for researchers who are interested in the development of semantic techniques in multiple resources. Moreover, it can help discover more information and relations that would not be discussed or evaluated in the work. Then, it tests and compares them with our discovered and evaluated relations (sibling and semantic similarity) to assess which can add more enhancement to the accuracy of the provided recommendations.

References

[1] M. Musen, B. Neumann, and R. Studer. *Intelligent Information Processing.* Springer, Berlin, Germany, 2002. (Cited on page 1.)

[2] S. Hartmann, H. Köhler, and J. Wang. Ontology Consolidation in Bioinformatics. In *APCCM '10:Proceedings of the Seventh Asia-Pacific Conference on Conceptual Modelling*, pages 15–22, Brisbane, Australia, 2010. (Cited on pages 6, 16, and 17.)

[3] O. Bodenreider and R. Stevens. Bio-ontologies: current trends and future directions. *Briefings in Bioinformatics*, 7(3):256–274, 2006. (Cited on page 6.)

[4] S. Maurer. *Handbook of Graph Theory*, chapter Directed acyclic graphs, pages 142–155. 2003. (Cited on pages 7 and 26.)

[5] Gene Ontology Consortium et al. The gene ontology (go) database and informatics resource. *Nucleic acids research*, 32(suppl 1):D258–D261, 2004. (Cited on page 7.)

[6] L. Ding, P. Kolari, Z. Ding, and S. Avancha. Using ontologies in the semantic web: A survey. *Ontologies: A Handbook of Principles, Concepts and Applications in Information Systems*, 14:79, 2007. (Cited on pages 14 and 15.)

[7] E. Miller. An introduction to the resource description framework. *Bulletin of the American Society for Information Science*, 25(1):15–19, 1998. (Cited on pages 14 and 15.)

[8] D. McGuinness, F. Van Harmelen, et al. Owl web ontology language overview. *W3C recommendation*, 10(10):2004, 2004. URL `http://www.w3.org/TR/owl-features/`. (Cited on pages 14, 16, and 42.)

[9] M. Cristani and R. Cuel. A survey on ontology creation methodologies. *International Journal on Semantic Web and Information Systems (IJSWIS)*, 1(2):49–69, 2005. (Cited on page 15.)

[10] E. Sirin, B. Parsia, B. Cuenca Grau, A. Kalyanpur, and Y. Katz. Pellet: A practical owl-dl reasoner. *Web Semantics: Science, Services and Agents on the World Wide Web*, 5(2):51 – 53, 2007. ISSN 1570-8268. (Cited on pages 16 and 33.)

[11] R. Shearer, B. Motik, and I. Horrocks. HermiT: A Highly-Efficient OWL Reasoner. In *OWLED '08: Proceedings of the 6th International Workshop on OWL: Experiences and Directions*, Karlsruhe, Germany, 2008. (Cited on pages 16 and 32.)

[12] J. Euzenat, P. Shvaiko, et al. *Ontology matching*, volume 18. Springer, 2007. (Cited on page 16.)

[13] D. Movshovitz-Attias, S. Whang, N. Noy, and A. Halevy. Discovering subsumption relationships for web-based ontologies. In *Proceedings of the 18th International Workshop on Web and Databases*, pages 62–69, Melbourne, VIC, Australia, 2015. ACM. (Cited on page 16.)

[14] C. Martinez-Cruz, C. Porcel, J. Bernabé-Moreno, and E. Herrera-Viedma. A model to represent users trust in recommender systems using ontologies and fuzzy linguistic modeling. *Information Sciences*, 311:102–118, 2015. (Cited on page 16.)

[15] S. Cardoso, F. Amanqui, K. Serique, J. dos Santos, and D. Moreira. Swi: A semantic web interactive gazetteer to support linked open data. *Future Generation Computer Systems*, 54:389–398, 2016. (Cited on page 16.)

[16] H. Müller, E. Kenny, and P. Sternberg. Textpresso: An ontology-based information retrieval and extraction system for biological literature. *PLoS Biology*, 2(11):1984–1998, 2004. (Cited on page 17.)

[17] L. Post, M. Roos, M. Marshall, R. van Driel, and T. Breit. A semantic web approach applied to integrative bioinformatics experimentation: a biological use case with genomics data. *Bioinformatics*, 23(22):3080–3087, 2007. (Cited on page 17.)

[18] R. Stevens, C. Goble, I. Horrocks, and S. Bechhofer. Building a bioinformatics ontology using oil. *Information Technology in Biomedicine, IEEE Transactions on*, 6(2):135–141, 2002. (Cited on page 17.)

[19] H. Menager, Z. Lacroix, and P. Tuffery. Bioinformatics Services Discovery Using Ontology Classification. In *SCC '13: Proceedings of the IEEE 9th World Congress on Services*, pages 106–113, Santa Clara Marriott, CA, 2007. (Cited on page 17.)

[20] N. Daraselia, A. Yuryev, S. Egorov, I. Mazo, and I. Ispolatov. Automatic extraction of gene ontology annotation and its correlation with clusters in protein networks. *BMC bioinformatics*, 8(1):243, 2007. (Cited on page 17.)

[21] H. González-Vélez. Guest editorial preface: Computational intelligence for neuro-oncological diagnosis. *The Knowledge Engineering Review*, 26:243–245, 7 2011. ISSN 1469-8005. (Cited on page 17.)

[22] Z. Aleksovski, W. ten Kate, and F. van Harmelen. Exploiting the Structure of Background Knowledge Used in Ontology Matching. In *OM '06: Proceedings of the 1st International Workshop on Ontology Matching*, Athens, Georgia, 2006. (Cited on page 17.)

[23] W. Blondé, V. Mironov, A. Venkatesan, E. Antezana, B. De Baets, and M. Kuiper. Reasoning with bio-ontologies: using relational closure rules to enable practical querying. *Bioinformatics*, 27(11):1562–1568, 2011. (Cited on page 17.)

[24] R. Foulger, P. Denny, J. Hardy, M. Martin, T. Sawford, and R. Lovering. Using the gene ontology to annotate key players in parkinson's disease. *Neuroinformatics*, 14:297–304, 2016. (Cited on page 17.)

[25] E. Galeota and M. Pelizzola. Ontology-based annotations and semantic relations in large-scale (epi)genomics data. *Briefings in Bioinformatics*, 2016. doi: 10.1093/bib/bbw036. (Cited on page 17.)

[26] F. Lekschas and N. Gehlenborg. Satori: A system for ontology-guided visual exploration of biomedical data repositories. *bioRxiv*, 2016. doi: 10.1101/046755. (Cited on page 18.)

[27] N. Noy. Ontology mapping. In *Handbook on ontologies*, pages 573–590. Springer, 2009. (Cited on page 18.)

[28] E. Chifu and I. Letia. A Neural Model for Ontology Matching. In *FedCSIS '11: Proceedings of the Federated Conference on Computer Science and Information Systems*, pages 933–940, Szczecin, Poland, 2011. (Cited on page 18.)

[29] N. Choi, I. Song, and H. Han. A survey on ontology mapping. *SIGMOD Rec.*, 35(3):34–41, September 2006. ISSN 0163-5808. doi: 10.1145/1168092.1168097. URL `http://doi.acm.org/10.1145/1168092.1168097`. (Cited on page 18.)

[30] Y. Ding and S. Foo. Ontology research and development. part 2-a review of ontology mapping and evolving. *Journal of information science*, 28(5):375–388, 2002. (Cited on page 18.)

[31] N. Arch-int and S. Arch-int. Semantic ontology mapping for interoperability of learning resource systems using a rule-based reasoning approach. *Expert Systems with Applications*, 40(18):7428–7443, 2013. (Cited on page 19.)

[32] C. Nuntawong, C. Namahoot, and M. Brückner. A semantic similarity assessment tool for computer science subjects using extended wu & palmer's algorithm and ontology. In *Information Science and Applications*, pages 989–996. Springer, 2015. (Cited on page 19.)

[33] Z. Wu and M. Palmer. Verbs semantics and lexical selection. In *ACL '94: Proceedings of the 32nd annual meeting on Association for Computational Linguistics*, pages 133–138, Las Cruces, New Mexico, 1994. (Cited on pages 19 and 25.)

[34] S. Kumar and J. Harding. Ontology mapping using description logic and bridging axioms. *Computers in Industry*, 64(1):19 – 28, 2013. ISSN 0166-3615. (Cited on page 19.)

[35] M. Hartung, A. Gross, T. Kirsten, and E. Rahm. Effective Mapping Composition for Biomedical Ontologies. In *SIMI '12: Proceedings of Semantic Interoperability in Medical Informatics*, pages 1–12, Heraklion, Greece, 2012. (Cited on page 20.)

[36] C. Knoblock, P. Szekely, J. Ambite, S. Gupta, A. Goel, M. Muslea, K. Lerman, and P. Mallick. Interactively mapping data sources into the semantic web. *LISC*, 783, 2011. (Cited on page 20.)

[37] M. Ehrig and Y. Sure. Ontology Mapping-An Integrated Approach. In *ESWS '04: Proceedings of 1st European Semantic Web Symposium*, pages 76–86, Heraklion, Greece, 2004. (Cited on page 21.)

[38] W. Kim, S. Park, S. Bang, and S. Lee. An Ontology Mapping Algorithm between Heterogeneous Product Classification Taxonomies. In *OM '06: Proceedings of the 1st International Workshop on Ontology Matching*, pages 347–360, Athens, Georgia, 2006. (Cited on page 21.)

[39] J. Li. Lom: a lexicon-based ontology mapping tool. Technical report, DTIC Document, 2004. (Cited on page 22.)

[40] G. Miller, R. Beckwith, C. Fellbaum, D. Gross, and K. Miller. Introduction to wordnet: An on-line lexical database*. *International journal of lexicography*, 3(4):235–244, 1990. (Cited on page 22.)

[41] A. Pease, I. Niles, and J. Li. The suggested upper merged ontology: A large ontology for the semantic web and its applications. In *AAAI '02:*

Proceedings of the Workshop on Ontologies and the Semantic Web, pages 25–39, Edmonton, Alberta, 2002. (Cited on page 22.)

[42] I. Niles and A. Terry. The MILO: A General-purpose, Mid-level Ontology. In *IKE '04: Proceedings of the International Conference on Information and Knowledge Engineering*, pages 15–19, Las Vegas, Nevada, 2004. (Cited on page 22.)

[43] S. Anam, Y. Kim, B. Kang, and Q. Liu. Adapting a knowledge-based schema matching system for ontology mapping. In *Proceedings of the Australasian Computer Science Week Multiconference*, ACSW '16, pages 27:1–27:10, Canberra, Australia, 2016. ACM. (Cited on page 22.)

[44] A. Khattak, Z. Pervez, W. Khan, A. Khan, K. Latif, and S. Lee. Mapping evolution of dynamic web ontologies. *Information Sciences*, 303:101 – 119, 2015. (Cited on page 24.)

[45] H. Al-Mubaid and H. Nguyen. Measuring semantic similarity between biomedical concepts within multiple ontologies. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 39(4): 389–398, 2009. (Cited on pages 25 and 28.)

[46] E. Blanchard, M. Harzallah, H. Bri, P. Kuntz, and R. Pauc. A Typology Of Ontology-Based Semantic Measures. In *EMOI-INTEROP '05: Proceedings of the Workshop on Enterprise Modelling and Ontologies for Interoperability*, pages 568–600, Porto, Portugal, 2005. (Cited on page 25.)

[47] P. Resnik. Semantic similarity in a taxonomy: An information-based measure and its application to problems of ambiguity in natural language. *Journal of Artificial Intelligence Research*, 11:95–130, 1999. (Cited on pages 25 and 26.)

[48] C. Leacock and M. Chodorow. Combining local context and wordnet similarity for word sense identification. *WordNet: An electronic lexical database*, 49(2):265–283, 1998. (Cited on page 25.)

[49] D. Lin. An Information-theoretic Definition of Similarity. In *ICML '98: Proceedings of the 15th International Conference on Machine Learning*, pages 296–304, Madison, Wisconsin, 1998. (Cited on pages 25, 26, and 125.)

[50] R. Rada, H. Mili, E. Bicknell, and M. Blettner. Development and application of a metric on semantic nets. *Systems, Man and Cybernetics, IEEE Transactions on*, 19(1):17–30, 1989. (Cited on page 25.)

[51] J. Jiang and D. Conrath. Semantic similarity based on corpus statistics and lexical taxonomy. *arXiv preprint cmp-lg/9709008*, cmp-lg-9709008, 1997. (Cited on pages 25 and 26.)

[52] M. Sussna. Word Sense Disambiguation for Free-text Indexing Using a Massive Semantic Network. In *CIKM '93: Proceedings of the 2nd International Conference on Information and Knowledge Management*, pages 67–74, Washington, DC, 1993. (Cited on page 25.)

[53] G. Hirst and D. St-Onge. Lexical chains as representations of context for the detection and correction of malapropisms. *WordNet: An electronic lexical database*, 305:305–332, 1998. (Cited on page 25.)

[54] Y. Li, Z. Bandar, and D. McLean. An approach for measuring semantic similarity between words using multiple information sources. *Knowledge and Data Engineering, IEEE Transactions on*, 15(4):871–882, 2003. (Cited on pages 25 and 32.)

[55] H. Yang, T. Nepusz, and A. Paccanaro. Improving go semantic similarity measures by exploring the ontology beneath the terms and modelling uncertainty. *Bioinformatics*, 28(10):1383–1389, 2012. (Cited on pages 26, 29, and 31.)

[56] C. Pesquita, D. Faria, H. Bastos, A. Ferreira, A. Falcão, and F. Couto. Metrics for go based protein semantic similarity: a systematic evaluation. *BMC Bioinformatics*, 9:1–16, 2008. (Cited on pages 26, 29, and 31.)

[57] F. Couto, M. Silva, and P. Coutinho. Measuring semantic similarity between gene ontology terms. *Data & knowledge engineering*, 61(1):137–152, 2007. (Cited on page 26.)

[58] F. Couto and M. Silva. Disjunctive shared information between ontology concepts: application to gene ontology. *Journal of Biomedical Semantics*, 2 (1):1–16, 2011. (Cited on page 26.)

[59] P. Lord, R. Stevens, A. Brass, and C. Goble. Investigating semantic similarity measures across the gene ontology: the relationship between sequence and annotation. *Bioinformatics*, 19(10):1275–1283, 2003. (Cited on page 26.)

[60] J. Sevilla, V. Segura, A. Podhorski, E. Guruceaga, J. Mato, L. Martínez-Cruz, F. Corrales, A. Rubio, et al. Correlation between gene expression and go semantic similarity. *Computational Biology and Bioinformatics, IEEE/ACM Transactions on*, 2(4):330–338, 2005. (Cited on page 26.)

[61] D. Sánchez, A. Solé-Ribalta, M. Batet, and F. Serratosa. Enabling semantic similarity estimation across multiple ontologies: An evaluation in the biomedical domain. *Journal of biomedical informatics*, 45(1):141–155, 2012. (Cited on page 27.)

[62] Z. Teng, M. Guo, X. Liu, Q. Dai, C. Wang, and P. Xuan. Measuring gene functional similarity based on group-wise comparison of go terms. *Bioinformatics*, 29:1424–1432, 2013. (Cited on pages 27, 67, 129, and 223.)

[63] O. Bodenreider. The unified medical language system (umls): integrating biomedical terminology. *Nucleic Acids Research*, 32(suppl 1):D267–D270, 2004. (Cited on page 28.)

[64] M. Alvarez Vega. *Graph Kernels and Applications in Bioinformatics.* PhD thesis, Utah State University, 2011. (Cited on page 29.)

[65] M. Alvarez, S. Lim, et al. A graph modeling of semantic similarity between words. In *ICSC '07: Proceedings of the 4th International Conference on Semantic Computing*, pages 355–362, Irvine, CA, 2007. (Cited on page 29.)

[66] D. Bollegala, Y. Matsuo, and M. Ishizuka. Measuring Semantic Similarity Between Words Using Web Search Engines. In *WWW '07: Proceedings of the 16th International Conference on World Wide Web*, pages 757–766, New York, NY, 2007. (Cited on page 29.)

[67] A. Nagar and H. Al-Mubaid. A hybrid semantic similarity measure for gene ontology based on offspring and path length. In *Computational Intelligence in Bioinformatics and Computational Biology (CIBCB), 2015 IEEE Conference on*, pages 1–7, Niagara Falls, ON, 2015. (Cited on page 30.)

[68] Q. Zhang and D. Haglin. Semantic similarity between ontologies at different scales. *IEEE/CAA Journal of Automatica Sinica*, 3(2):132–140, 2016. (Cited on page 31.)

[69] R. Thiagarajan, G. Manjunath, and M. Stumptner. Computing Semantic Similarity Using Ontologies. In *ISWC '08: Proceedings of the International Semantic Web Conference*, Karlsruhe, Germany, 2008. (Cited on page 32.)

[70] L. Badea, D. Tilivea, and A. Hotaran. Semantic web reasoning for ontology-based integration of resources. In *Principles and Practice of Semantic Web Reasoning*, pages 61–75. Springer, 2004. (Cited on page 32.)

[71] J. Hastings, M. Dumontier, D. Hull, M. Horridge, C. Steinbeck, U. Sattler, R. Stevens, T. Horne, and K. Britz. Representing chemicals using owl, description graphs and rules. 2010. (Cited on page 32.)

[72] B. DuCharme. *Learning SPARQL.* O'Reilly Media, Inc., 2011. ISBN 1449306594, 9781449306595. (Cited on pages 33 and 134.)

[73] M. Luther, T. Liebig, S. Böhm, and O. Noppens. Who the heck is the father of bob? In *The Semantic Web: Research and Applications*, volume 5554, pages 66–80. Springer, 2009. (Cited on page 33.)

[74] K. Okoye, A. Tawil, U. Naeem, and E. Lamine. *Advances in Nature and Biologically Inspired Computing: Proceedings of the 7th World Congress on Nature and Biologically Inspired Computing (NaBIC2015) in Pietermaritzburg, South Africa, held December 01-03, 2015*, chapter A Semantic Reasoning

Method Towards Ontological Model for Automated Learning Analysis, pages 49–60. Springer International Publishing, 2016. (Cited on page 33.)

[75] M. Torres, E. Loza, W. Al-Halabi, G. Guzman, R. Quintero, and M. Moreno-Ibarra. Qualitative spatial reasoning methodology to determine the particular domain of a set of geographic objects. *Computers in Human Behavior*, 59:115 – 133, 2016. ISSN 0747-5632. (Cited on page 33.)

[76] H. Jamalabadi, H. Nasrollahi, S. Alizadeh, B. Araabi, and M. Ahamadabadi. Competitive interaction reasoning: A bio-inspired reasoning method for fuzzy rule based classification systems. *Information Sciences*, 352-353:35 – 47, 2016. (Cited on page 33.)

[77] Z. Huang and F. Harmelen. Using Semantic Distances for Reasoning with Inconsistent Ontologies. In *ISWC '08: Proceedings of the 7th International Conference on The Semantic Web*, pages 178–194, Karlsruhe, Germany, 2008. (Cited on page 34.)

[78] Z. Zhong, Z. Liu, C. Li, and Y. Guan. Event ontology reasoning based on event class influence factors. *International Journal of Machine Learning and Cybernetics*, 3(2):133–139, 2012. (Cited on page 34.)

[79] J. Fang, Z. Huang, and F. van Harmelen. A Method of Contrastive Reasoning with Inconsistent Ontologies. In *JIST '11: Proceedings of the 2011 Joint International Conference on The Semantic Web*, pages 1–16, Hangzhou, China, 2012. (Cited on page 35.)

[80] L. Serafini and A. Tamilin. DRAGO: Distributed Reasoning Architecture for the Semantic Web. In *ESWC '05: Proceedings of the 2nd European Conference on The Semantic Web: Research and Applications*, pages 361–376, Heraklion, Greece, 2005. (Cited on page 35.)

[81] D. Elenius, D. Martin, R. Ford, and G. Denker. Reasoning About Resources and Hierarchical Tasks Using OWL and SWRL. In *ISWC '09: Proceedings of the 8th International Semantic Web Conference*, pages 795–810, Chantilly, VA, 2009. (Cited on pages 36, 37, 67, 129, 222, and 223.)

[82] B. Parsiaa, E. Sirinb, B. Graua, E. Ruckhausa, and D. Hewlettb. Cautiously approaching swrl. Technical report, University of Maryland, 2005. (Cited on pages 37 and 107.)

[83] D. Radev and K. McKeown. Generating natural language summaries from multiple on-line sources. *Comput. Linguist.*, 24(3):470–500, September 1998. ISSN 0891-2017. URL `http://dl.acm.org/citation.cfm?id=972749.972755`. (Cited on page 37.)

[84] S. Pereira Detro, D. Morozov, M. Lezoche, H. Panetto, E. Portela Santos, and M. Zdravkovic. Enhancing semantic interoperability in healthcare using semantic process mining. In *6th International Conference on Information Society and Techology, ICIST 2016*, volume 1, pages 80–85, Kopaonik, Serbia, 2016. ISBN: 978-86-85525-18-6. (Cited on page 38.)

[85] A. Rakib, A. Lawan, and S. Walker. *An Ontological Approach for Knowledge Modeling and Reasoning Over Heterogeneous Crop Data Sources*, pages 35–47. Springer International Publishing, 2015. (Cited on pages 39, 67, 129, 222, and 223.)

[86] L. Serafini and A. Tamilin. Distributed reasoning services for multiple ontologies. Technical report, University of Trento, 2004. (Cited on page 39.)

[87] D. Kim, K. Barker, and B. Porter. Knowledge Integration Across Multiple Texts. In *K-CAP '09: Proceedings of the Fifth International Conference on Knowledge Capture*, pages 49–56, Redondo Beach, CA, 2009. (Cited on page 39.)

[88] P. Rangel, J. Junior, M. Ramirez, and J. de Souza. Context Reasoning Through a Multiple Logic Framework. In *IE '10: Proceedings of the 2010 6th International Conference on Intelligent Environments*, pages 116–121, Kuala Lumpur, Malaysia, 2010. (Cited on page 39.)

[89] J. Lu, X. Sun, L. Xu, and H. Wang. Incremental Reasoning over Multiple Ontologies. In *WAIM '11: Proceedings of the 12th International Conference on Web-age Information Management*, pages 131–143, Wuhan, China, 2011. (Cited on page 39.)

[90] P. Bouché, C. Zanni-Merk, N. Gartiser, D. Renaud, and F. Rousselot. Reasoning with Multiple Points of View: A Case Study. In *KES'10: Proceedings of the 14th International Conference on Knowledge-based and Intelligent Information and Engineering Systems*, pages 32–40, Cardiff, UK, 2010. (Cited on page 39.)

[91] T. Kirayama and T. Tomiyama. Reasoning about Models across Multiple Ontologies. In *Proceedings of the International Qualitative Reasoning Workshop*, pages 124–131, Washington, DC, 1993. (Cited on page 39.)

[92] K. Kaneiwa and R. Mizoguchi. Distributed reasoning with ontologies and rules in order-sorted logic programming. *Web Semantics: Science, Services and Agents on the World Wide Web*, 7(3):252 – 270, 2009. ISSN 1570-8268. doi: http://dx.doi.org/10.1016/j.websem.2009.05.003. URL `http://www.sciencedirect.com/science/article/pii/S1570826809000183`. (Cited on page 39.)

[93] H. Chen, X. Chen, P. Gu, Z. Wu, and T. Yu. Owl reasoning framework over big biological knowledge network. *BioMed research international*, 2014: 272915, 2014. ISSN 2314-6133. doi: 10.1155/2014/272915. URL `http://europepmc.org/articles/PMC4022201`. (Cited on page 40.)

[94] J. Dean and S. Ghemawat. Mapreduce: Simplified data processing on large clusters. *Commun. ACM*, 51(1):107–113, January 2008. ISSN 0001-0782. (Cited on page 40.)

[95] J. Wang, D. Crawl, I. Altintas, K. Tzoumas, and V. Markl. Comparison of distributed data-parallelization patterns for big data analysis: A bioinformatics case study. In *DataCloud '13: Proceedings of the 4th International Workshop on Data Intensive Computing in the Clouds*, Chicago, Illinois, 2013. (Cited on page 41.)

[96] L. Tari, N. Vo, S. Liang, J. Patel, C. Baral, and J. Cai. Identifying novel drug indications through automated reasoning. *PloS one*, 7(7):e40946, 2012. (Cited on page 41.)

[97] G. Tsafnat and E. Coiera. Computational reasoning across multiple models. *Journal of the American Medical Informatics Association: JAMIA*, 16(6): 768, 2009. (Cited on page 41.)

[98] R. Hoehndorf, M. Dumontier, A. Oellrich, D. Rebholz-Schuhmann, P. Schofield, and G. Gkoutos. Interoperability between biomedical ontologies through relation expansion, upper-level ontologies and automatic reasoning. *PLoS One*, 6(7):e22006, 2011. (Cited on pages 42, 67, and 222.)

[99] R. Hoehndorf. What is an upper level ontology? `http://ontogenesis.knowledgeblog.org/740`, 2010. URL `http://ontogenesis.knowledgeblog.org/740`. (Cited on page 42.)

[100] M. Samwald, J. Giménez, R. Boyce, R. Freimuth, K. Adlassnig, and M. Dumontier. Pharmacogenomic knowledge representation, reasoning and genome-based clinical decision support based on owl 2 dl ontologies. *BMC medical informatics and decision making*, 15(1):12, 2015. (Cited on pages 42, 67, and 222.)

[101] I. Mallona, M. Jordà, and M. Peinado. A knowledgebase of the human alu repetitive elements. *Journal of Biomedical Informatics*, 60:77 – 83, 2016. (Cited on page 43.)

[102] Y. Zhang, H. Wu, J. Du, J. Xu, J. Wang, C. Tao, L. Li, and H. Xu. Extracting drug-enzyme relation from literature as evidence for drug drug interaction. *Journal of Biomedical Semantics*, 7(1):1–8, 2016. (Cited on page 44.)

[103] S. Kohler, S. Bauer, C. Mungall, G. Carletti, C. Smith, P. Schofield, G. Gkoutos, and P. Robinson. Improving ontologies by automatic reasoning and evaluation of logical definitions. *BMC Bioinformatics*, 12(1):418, 2011. ISSN 1471-2105. (Cited on page 45.)

[104] F. Baader, C. Lutz, and B. Suntisrivaraporn. CEL: A Polynomial-time Reasoner for Life Science Ontologies. In *IJCAR '06: Proceedings of the 3rd International Joint Conference on Automated Reasoning*, pages 287–291, Seattle, WA, 2006. (Cited on page 45.)

[105] E. Horvitz. *Automated reasoning for biology and medicine.* Knowledge Systems Laboratory, Section on Medical Informatics, Stanford University, 1992. (Cited on page 45.)

[106] M. Mezghani, C. Zayani, I. Amous, and F. Gargouri. A User Profile Modelling Using Social Annotations: A Survey. In *WWW '12: Proceedings of the 21st International Conference Companion on World Wide Web*, pages 969–976, Lyon, France, 2012. (Cited on pages 45 and 47.)

[107] C. Zayani. *Contribution to the definition and implementation of adaptation mechanisms of semi-structured documents.* PhD thesis, University of Toulouse, University Paul Sabatier Toulouse III-2008, 2008. (Cited on page 45.)

[108] B. Mobasher. Data mining for web personalization. In *The adaptive web*, pages 90–135. Springer, 2007. (Cited on page 45.)

[109] D. Morikawa, M. Honjo, A. Yamaguchi, and M. Ohashi. Profile Aggregation and Dissemination: A Framework for Personalized Service Provisioning. In *International Workshop on Advanced Context Modelling, Reasoning and Management*, Tokyo, Japan, 2004. (Cited on page 45.)

[110] Y. Kritikou, P. Demestichas, E. Adamopoulou, K. Demestichas, M. Theologou, and M. Paradia. User profile modeling in the context of web-based learning management systems. *Journal of Network and Computer Applications*, 31(4):603 – 627, 2008. ISSN 1084-8045. doi: http://dx.doi.org/10.1016/j.jnca.2007.11.006. URL `http://www.sciencedirect.com/science/article/pii/S1084804507000720`. (Cited on page 45.)

[111] S. Chaudhuri and A. Tewari. Online Learning to Rank with Feedback at the Top. In *The 19th International Conference on Artificial Intelligence and Statistics (AISTATS 2016)*, pages 277–285, Cadiz, Spain, 2016. (Cited on page 46.)

[112] A. Nwana and T. Chen. Towards understanding user preferences from user tagging behavior for personalization. In *2015 IEEE International Symposium*

on Multimedia (ISM), pages 178–183, Mimai,FL, 2015. IEEE. (Cited on page 46.)

[113] J. Trajkova and S. Gauch. Improving Ontology-Based User Profiles. In *RIAO '04: Proceedings of the 7th International Conference on Computer-Assisted Information Retrieval*, pages 380–390, Avignon, France, 2004. (Cited on pages 46 and 55.)

[114] M. Daoud, L. Tamine-Lechani, and M. Boughanem. Learning User Interests for a Session-based Personalized Search. In *IIiX '08: Proceedings of the 2nd International Symposium on Information Interaction in Context*, pages 57–64, London, UK, 2008. (Cited on pages 46 and 52.)

[115] J. Wang, B. Sarwar, and N. Sundaresan. Utilizing Related Products for Post-purchase Recommendation in e-Commerce. In *RecSys '11: Proceedings of the 5th ACM Conference on Recommender Systems*, pages 329–332, Chicago, Illinois, 2011. (Cited on page 46.)

[116] M. Najafabadi and M. Mahrin. A systematic literature review on the state of research and practice of collaborative filtering technique and implicit feedback. *Artificial Intelligence Review*, 45(2):167–201, 2016. (Cited on page 47.)

[117] L. Anh-Thu, H. Nguyen, and N. Thai-Nghe. *A Context-Aware Implicit Feedback Approach for Online Shopping Recommender Systems*, pages 584–593. Springer Berlin Heidelberg, 2016. (Cited on page 47.)

[118] B. Dangra, M. Bedekar, and S. Panicker. User profiling of automobile driver and outlier detection. *International Journal of Innovative Research and Development*, 3(12), 2014. (Cited on page 47.)

[119] G. Amato and U. Straccia. *User Profile Modeling and Applications to Digital Libraries*, pages 184–197. Springer Berlin Heidelberg, Berlin, Heidelberg, 1999. (Cited on page 47.)

[120] B. Cornelis. Personalizing search in digital libraries. Master's thesis, Universiteit Maastricht, 2003. (Cited on page 47.)

[121] Definition of user profile in english. http://www.oxforddictionaries.com/definition/english/user-profile?q=user+profile. Accessed: 14-06-2016. (Cited on page 48.)

[122] Y. Webster. *A Hybrid Approach for Translational Research.* PhD thesis, School of Informatics, Indiana University, 2010. (Cited on pages 48, 55, 67, and 223.)

[123] S. Gauch, M. Speretta, A. Chandramouli, and A. Micarelli. User profiles for personalized information access. In *The adaptive web*, pages 54–89. Springer, 2007. (Cited on pages 48, 49, and 50.)

[124] G. Salton and M. McGill. *Introduction to Modern Information Retrieval.* McGraw-Hill, Inc., New York, NY, USA, 1986. ISBN 0070544840. (Cited on pages 49 and 83.)

[125] K. Jones. A statistical interpretation of term specificity and its application in retrieval. *Journal of documentation*, 28(1):11–21, 1972. (Cited on pages 49, 81, and 82.)

[126] M. Alaofi and G. Rumantir. Personalisation of generic library search results using student enrolment information. *JEDM-Journal of Educational Data Mining*, 7(3):68–88, 2015. (Cited on page 49.)

[127] G. Gentili, A. Micarelli, and F. Sciarrone. Infoweb: An adaptive information filtering system for the cultural heritage domain. *Applied Artificial Intelligence*, 17(8-9):715–744, 2003. (Cited on page 49.)

[128] K. Lakiotaki, A. Hliaoutakis, S. Koutsos, and E. Petrakis. *Towards Personalized Medical Document Classification by Leveraging UMLS Semantic Network*, pages 93–104. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013. (Cited on page 49.)

[129] K. Skillen, L. Chen, C. Nugent, M. Donnelly, W. Burns, and I. Solheim. Ontological User Profile Modeling for Context-aware Application Personalization. In *UCAmI'12: Proceedings of the 6th International Conference*

on Ubiquitous Computing and Ambient Intelligence, pages 261–268, Vitoria-Gasteiz, Spain, 2012. (Cited on page 50.)

[130] M. Reformat and S. Golmohammadi. Rule- and owa-based semantic similarity for user profiling. *International Journal of Fuzzy Systems*, 12(2):87–102, 2010. (Cited on page 50.)

[131] Z. Yu, Y. Nakamura, S. Jang, S. Kajita, and K. Mase. Ontology-based Semantic Recommendation for Context-aware e-Learning. In *UIC '07: Proceedings of the 4th International Conference on Ubiquitous Intelligence and Computing*, pages 898–907, Hong Kong, China, 2007. (Cited on page 50.)

[132] V. Luna, R. Quintero, M. Torres, M. Moreno-Ibarra, G. Guzmán, and I. Escamilla. An ontology-based approach for representing the interaction process between user profile and its context for collaborative learning environments. *Computers in Human Behavior*, 51, Part B:1387 – 1394, 2015. ISSN 0747-5632. Computing for Human Learning, Behaviour and Collaboration in the Social and Mobile Networks Era. (Cited on page 50.)

[133] J. Srivastava, R. Cooley, M. Deshpande, and P. Tan. Web usage mining: Discovery and applications of usage patterns from web data. *SIGKDD Explor. Newsl.*, 1(2):12–23, January 2000. ISSN 1931-0145. doi: 10.1145/846183.846188. URL `http://doi.acm.org/10.1145/846183.846188`. (Cited on page 50.)

[134] C. Wong, S. Shiu, and S. Pal. Mining Fuzzy Association Rules for Web Access Case Adaptation. In *Proceedings of Soft Computing in Case-Based Reasoning Workshop, in conjunction with the 4th International Conference in Case-Based Reasoning*, pages 213–220, Vancouver, Canada, 2001. (Cited on page 50.)

[135] Y. Wakita, K. Oku, H. Huang, and K. Kawagoe. A fashion-brand recommender system using brand association rules and features. In *Advanced Applied Informatics (IIAI-AAI), 2015 IIAI 4th International Congress on*, pages 719–720, Okayama, Japan, 2015. IEEE. (Cited on page 50.)

[136] J. He. *A Social Network-based Recommender System.* PhD thesis, Los Angeles, CA, USA, 2010. AAI3437557. (Cited on page 51.)

[137] M. Pazzani and D. Billsus. Content-based recommendation systems. In *The adaptive web*, pages 325–341. Springer, 2007. (Cited on pages 51 and 53.)

[138] R. Mooney and L. Roy. Content-based Book Recommending Using Learning for Text Categorization. In *DL '00: Proceedings of the 5th ACM Conference on Digital Libraries*, pages 195–204, San Antonio, Texas, 2000. (Cited on page 51.)

[139] H. Alharthi and D. Inkpen. *Content-Based Recommender System Enriched with Wordnet Synsets*, pages 295–308. Springer International Publishing, 2015. (Cited on page 51.)

[140] M. Fasli. *Agent Technology For E-Commerce.* John Wiley & Sons, 2007. ISBN 0470030305. (Cited on page 51.)

[141] B. Sarwar, J. Konstan, A. Borchers, J. Herlocker, B. Miller, and J. Riedl. Using Filtering Agents to Improve Prediction Quality in the GroupLens Research Collaborative Filtering System. In *CSCW '98: Proceedings of the 1998 ACM Conference on Computer Supported Cooperative Work*, pages 345–354, Seattle, Washington, 1998. (Cited on page 51.)

[142] A. Al-Badarenah and J. Alsakran. An automated recommender system for course selection. *International Journal of Advanced Computer Science & Applications*, 7(3):166–175, 2016. (Cited on page 51.)

[143] Y. Freund, R. Iyer, R. Schapire, and Y. Singer. An efficient boosting algorithm for combining preferences. *J. Mach. Learn. Res.*, 4:933–969, December 2003. ISSN 1532-4435. URL `http://dl.acm.org/citation.cfm?id=945365.964285`. (Cited on page 51.)

[144] P. Melville, R. Mooney, and R. Nagarajan. Content-boosted collaborative filtering for improved recommendations. In *AAAI '02: Proceedings of the 18th National Conference on Artificial Intelligence*, pages 187–192, Edmonton, Alberta, 2002. (Cited on page 52.)

[145] J. Hao, Y. Yan, G. Wang, L. Gong, and B. Zhao. A probability-based hybrid user model for recommendation system. *Mathematical Problems in Engineering*, 2016, 2016. (Cited on page 52.)

[146] T. Vu, A. Willis, S. Tran, and D. Song. *Temporal Latent Topic User Profiles for Search Personalisation*, pages 605–616. Springer International Publishing, 2015. (Cited on page 53.)

[147] A. Micarelli, F. Gasparetti, F. Sciarrone, and S. Gauch. The adaptive web. chapter Personalized Search on the World Wide Web, pages 195–230. Springer-Verlag, Berlin, Heidelberg, 2007. ISBN 978-3-540-72078-2. URL `http://dl.acm.org/citation.cfm?id=1768197.1768205`. (Cited on page 53.)

[148] R. Swezey, S. Shiramatsu, T. Ozono, and T. Shintani. Intelligent Page Recommender Agents: Real-time Content Delivery for Articles and Pages Related to Similar Topics. In *IEA/AIE '11: Proceedings of the 24th International Conference on Industrial Engineering and Other Applications of Applied Intelligent Systems Conference on Modern Approaches in Applied Intelligence*, pages 173–182, Syracuse, NY, 2011. (Cited on page 53.)

[149] Y. Blanco-Fernandez, J. Pazos-Arias, A. Gil-Solla, M. Ramos-Cabrer, B. Barragans-Martinez, and M. Lopez-Nores. A Multi-Agent Open Architecture for a TV Recommender System: A Case Study Using a Bayesian Strategy. In *ISMSE '04: Proceedings of the IEEE 6th International Symposium on Multimedia Software Engineering*, pages 178–185, Miami, Florida, 2004. (Cited on page 54.)

[150] W. Cui. A chinese text classification system based on naive bayes algorithm. *MATEC Web of Conferences*, 44:01015, 2016. (Cited on page 54.)

[151] C. Shahabi and Y. Chen. An adaptive recommendation system without explicit acquisition of user relevance feedback. *Distributed and Parallel Databases*, 14(2):173–192, 2003. (Cited on page 55.)

[152] C. Shahabi, F. Kashani, Y. Chen, and D. McLeod. Yoda: An Accurate and Scalable Web-Based Recommendation System. In *CooplS '01: Proceedings of*

the 9th International Conference on Cooperative Information Systems, pages 418–432, London, UK, 2001. (Cited on page 55.)

[153] V. Lavrenko and W. Croft. Relevance Based Language Models. In *SIGIR '01: Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 120–127, New Orleans, Louisiana, 2001. (Cited on page 55.)

[154] S. Sheth, J. Bell, N. Arora, and G. Kaiser. Towards diversity in recommendations using social networks. Technical report, 2011. (Cited on page 56.)

[155] T. Yoneya and H. Mamitsuka. PURE: a PubMed article recommendation system based on content-based filtering. In *IBSB '07: Proceedings of The Seventh Annual International Workshop on Bioinformatics and Systems Biology*, pages 267–276, Tokyo, Japan, 2007. (Cited on page 57.)

[156] S. Middleton, N. Shadbolt, and D. De Roure. Ontological user profiling in recommender systems. *ACM Trans. Inf. Syst.*, 22(1):54–88, January 2004. ISSN 1046-8188. doi: 10.1145/963770.963773. URL `http://doi.acm.org/10.1145/963770.963773`. (Cited on page 57.)

[157] S. Middleton, D. De Roure, and N. Shadbolt. Ontology-based recommender systems. In *Handbook on ontologies*, pages 477–498. Springer, 2004. (Cited on page 57.)

[158] S. Middleton, D. De Roure, and N. Shadbolt. Ontology-based recommender systems. In *Handbook on ontologies*, pages 779–796. Springer, 2009. (Cited on page 57.)

[159] R. Mirizzi, T. Di Noia, V. Ostuni, and A. Ragone. Linked open data for content-based recommender systems. Technical report, http://sisinflab.poliba.it/semantic-expert-finding/papers/tech-report-1-2012.pdf, 2012. (Cited on pages 58, 67, 96, 163, 166, 167, 169, 170, 188, 189, 190, 192, 193, 214, 218, 224, 225, and 280.)

[160] G. Salton, A. Wong, and C. Yang. A vector space model for automatic indexing. *Commun. ACM*, 18(11):613–620, November 1975. ISSN 0001-

0782. doi: 10.1145/361219.361220. URL `http://doi.acm.org/10.1145/361219.361220`. (Cited on pages 58 and 166.)

[161] T. Di Noia, R. Mirizzi, V. Ostuni, D. Romito, and M. Zanker. Linked Open Data to Support Content-based Recommender Systems. In *I-SEMANTICS '12: Proceedings of the 8th International Conference on Semantic Systems*, pages 1–8, Graz, Austria, 2012. (Cited on pages 58, 67, 224, and 225.)

[162] J. Ge, Z. Chen, J. Peng, and T. Li. An Ontology-Based Method for Personalized Recommendation. In *ICCI '12: Proceedings of the 11th International Conference on Cognitive Informatics and Cognitive Computing*, pages 522–526, Kyoto, Japan, 2012. (Cited on page 59.)

[163] Y. Shen, J. Yu, and K. Nan. A Hybrid Recommender Model for Scientific Research Resources. In *WiCOM '12: Proceedings of the 8th International Conference on Wireless Communications, Networking and Mobile Computing*, pages 1–4, Shanghai, China, 2012. (Cited on page 59.)

[164] R. Meymandpour and J. Davis. Enhancing Recommender Systems Using Linked Open Data-Based Semantic Analysis of Items. In *AWC '15: Proceedings of the 3rd Australasian Web Conference*, pages 30–41, Sydney, Australia, 2015. (Cited on pages 59, 67, 224, and 291.)

[165] D. Bianchini, V. De Antonellis, and M. Melchiori. *A Food Recommendation System Based on Semantic Annotations and Reference Prescriptions*, pages 134–143. Springer International Publishing, 2015. (Cited on page 60.)

[166] T. Erekhinskaya, M. Balakrishna, M. Tatu, and D. Moldovan. *Personalized Medical Reading Recommendation: Deep Semantic Approach*, pages 89–97. Springer International Publishing, 2016. (Cited on page 61.)

[167] K. Verbert, H. Drachsler, N. Manouselis, M. Wolpers, R. Vuorikari, and E. Duval. Dataset-driven Research for Improving Recommender Systems for Learning. In *LAK '11: Proceedings of the 1st International Conference on Learning Analytics and Knowledge*, pages 44–53, Alberta, Canada, 2011. (Cited on page 62.)

[168] L. Zhuhadar and O. Nasraoui. Personalized Search Based on a User-Centered Recommender Engine. In *WI-IAT '10: Proceedings of the International Conference on Web Intelligence and Intelligent Agent Technology*, pages 200–203, Toronto, Canada, 2010. (Cited on page 62.)

[169] M. Hameed, M. Malik, S. Sayeedunnisa, and H. Imroze. An Effective Hybrid Algorithm in Recommender Systems Based on Fast Genetic K-means and Information Gain. In *CICN '12: Proceedings of the 4th International Conference on Computational Intelligence and Communication Networks*, pages 860–865, Uttar Pradesh, India, 2012. (Cited on page 62.)

[170] O. McBryan. Genvl and wwww: Tools for taming the web. In *Proceedings of the first international world wide web conference*, volume 341, Geneva, Switzerland, 1994. (Cited on page 62.)

[171] S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine* 1. *Computer networks and ISDN systems*, 30(1-7):107–117, 1998. (Cited on page 62.)

[172] D. Goldberg, D. Nichols, B. Oki, and D. Terry. Using collaborative filtering to weave an information tapestry. *Communications of the ACM*, 35(12): 61–70, 1992. (Cited on page 62.)

[173] Y. Hu, Y. Koren, and C. Volinsky. Collaborative filtering for implicit feedback datasets. In *2008 Eighth IEEE International Conference on Data Mining*, pages 263–272, pisa,Italy, 2008. (Cited on page 62.)

[174] J. Edmonds, L. Raschid, H. Sayyadi, and S. Wu. Exploiting social media to provide humanitarian users with event search and recommendations. In *Proceedings of the 7th International Conference on Information Systems for Crisis Response and Management ISCRAM2010. Seattle, USA*, Seattle, USA. (Cited on page 63.)

[175] S. Vargas and P. Castells. Exploiting the diversity of user preferences for recommendation. In *Proceedings of the 10th Conference on Open Research*

Areas in Information Retrieval, OAIR '13, pages 129–136, Lisbon, Portugal, 2013. LE CENTRE DE HAUTES ETUDES INTERNATIONALES D'INFORMATIQUE DOCUMENTAIRE. (Cited on page 63.)

[176] H. Wang, S. Shao, X. Zhou, C. Wan, and A. Bouguettaya. Preference recommendation for personalized search. *Knowledge-Based Systems*, 100:124 – 136, 2016. (Cited on page 63.)

[177] G. Kim, K. Park, and D. Lee. A semantic-based health advising system exploiting web-based personal health record services. In *Computer Software and Applications Conference (COMPSAC), 2015 IEEE 39th Annual*, volume 3, pages 654–655, Taichung, Taiwan, 2015. (Cited on pages 63 and 282.)

[178] A. Livne, V. Gokuladas, J. Teevan, S. Dumais, and E. Adar. Citesight: Supporting contextual citation recommendation using differential search. In *Proceedings of the 37th International ACM SIGIR Conference on Research &#38; Development in Information Retrieval*, SIGIR '14, pages 807–816, Gold Coast, Queensland, Australia, 2014. (Cited on pages 64 and 287.)

[179] Z. Saaya, M. Schaal, M. Coyle, P. Briggs, and B. Smyth. *Exploiting Extended Search Sessions for Recommending Search Experiences in the Social Web*, pages 369–383. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012. (Cited on page 65.)

[180] Z. Saaya, B. Smyth, M. Coyle, and P. Briggs. *Recommending Case Bases: Applications in Social Web Search*, pages 274–288. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011. (Cited on page 65.)

[181] J. Lehmann, R. Isele, M. Jakob, A. Jentzsch, D. Kontokostas, P. Mendes, S. Hellmann, M. Morsey, P. van Kleef, S. Auer, et al. Dbpedia–a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic Web*, 5: 1–29, 2014. (Cited on pages 74 and 75.)

[182] S. Auer and J. Lehmann. What Have Innsbruck and Leipzig in Common? Extracting Semantics from Wiki Content. In *ESWC '07: Proceedings of the 4th European Conference on The Semantic Web: Research and Applications*, pages 503–517, Innsbruck, Austria, 2007. (Cited on page 74.)

[183] J. Chiu, T. Lee, S. Lee, H. Zhu, and D. Cheung. Extraction of rdf dataset from wikipedia infobox data. Technical report, Tech. rep., Department of Computer Science, The University of Hong Kong, 2010. (Cited on page 74.)

[184] M. Völkel, M. Krötzsch, D. Vrandecic, H. Haller, and R. Studer. Semantic Wikipedia. In *WWW '06: Proceedings of the 15th International Conference on World Wide Web*, pages 585–594, Edinburgh, UK, 2006. (Cited on page 74.)

[185] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, and Z. Ives. DBpedia: A Nucleus for a Web of Open Data. In Karl Aberer, Key-Sun Choi, Natasha Noy, Dean Allemang, Kyung-Il Lee, Lyndon Nixon, Jennifer Golbeck, Peter Mika, Diana Maynard, Riichiro Mizoguchi, Guus Schreiber, and Philippe Cudré-Mauroux, editors, *The Semantic Web*, volume 4825 of *Lecture Notes in Computer Science*, chapter 52, pages 722–735. Springer Berlin Heidelberg, 2007. ISBN 978-3-540-76297-3. doi: 10.1007/978-3-540-76298-0_52. URL `http://dx.doi.org/10.1007/978-3-540-76298-0_52`. (Cited on page 74.)

[186] P. Exner and P. Nugues. Entity extraction: From unstructured text to DBpedia RDF triples. In *WoLE '12: Proceedings of the Web of Linked Entities Workshop*, Boston, USA, 2012. (Cited on page 74.)

[187] A. Passant. Dbrec: Music Recommendations Using DBpedia. In *ISWC '10: Proceedings of the 9th International Semantic Web Conference on The Semantic Web*, pages 209–224, Shanghai, China, 2010. (Cited on page 74.)

[188] G. Kobilarov, T. Scott, Y. Raimond, S. Oliver, C. Sizemore, M. Smethurst, C. Bizer, and R. Lee. Media meets semantic web–how the bbc uses dbpedia and linked data to make connections. In *The semantic web: research and applications*, pages 723–737. Springer, 2009. (Cited on page 75.)

[189] G. Shani and A. Gunawardana. Evaluating recommendation systems. In *Recommender systems handbook*, pages 257–297. Springer, 2011. (Cited on pages 87, 91, 162, 191, and 218.)

[190] J. Hannon, M. Bennett, and B. Smyth. Recommending Twitter Users to Follow Using Content and Collaborative Filtering Approaches. In *RecSys '10: Proceedings of the 4th ACM Conference on Recommender Systems*, pages 199–206, Barcelona, Spain, 2010. (Cited on page 87.)

[191] S. McNee, I. Albert, D. Cosley, P. Gopalkrishnan, S. Lam, A. Rashid, J. Konstan, and J. Riedl. On the Recommending of Citations for Research Papers. In *CSCW '02: Proceedings of the 2002 ACM Conference on Computer Supported Cooperative Work*, pages 116–125, New Orleans, Louisiana, 2002. (Cited on page 87.)

[192] P. Pu, L. Chen, and R. Hu. A User-centric Evaluation Framework for Recommender Systems. In *RecSys '11: Proceedings of the 5th ACM Conference on Recommender Systems*, pages 157–164, Chicago, Illinois, 2011. (Cited on page 87.)

[193] B. Knijnenburg, M. Willemsen, and A. Kobsa. A Pragmatic Procedure to Support the User-centric Evaluation of Recommender Systems. In *RecSys '11: Proceedings of the 5th ACM Conference on Recommender Systems*, pages 321–324, Chicago, Illinois, 2011. (Cited on pages 87, 168, 169, and 193.)

[194] C. Mulwa, L. Longo, S. Lawless, M. Sharp, and V. Wade. An Online Framework for Supporting the Evaluation of Personalised Information Retrieval Systems. In *iUBICOM'11: Proceedings of the 6th International Conference on Ubiquitous and Collaborative Computing*, pages 75–85, Newcastle, UK, 2011. (Cited on page 87.)

[195] L. Chen and P. Pu. User evaluation framework of recommender systems. In *SRS '10: Proceedings of the Workshop on Social Recommender Systems*, Hong Kong, China, 2010. (Cited on page 87.)

[196] C. Hayes and P. Cunningham. An on-line evaluation framework for recommender systems. Technical report, Trinity College Dublin, Department of Computer Science, 2002. (Cited on page 88.)

[197] R. Kohavi, R. Longbotham, D. Sommerfield, and R. Henne. Controlled experiments on the web: survey and practical guide. *Data Mining and Knowledge Discovery*, 18(1):140–181, 2009. (Cited on page 88.)

[198] L. Marinho, A. Hotho, R. Jäschke, A. Nanopoulos, S. Rendle, L. Schmidt-Thieme, G. Stumme, and P. Symeonidis. *Recommender Systems for Social Tagging Systems*. Springer, 2012. (Cited on page 88.)

[199] A. Gunawardana and G. Shani. A survey of accuracy evaluation metrics of recommendation tasks. *J. Mach. Learn. Res.*, 10:2935–2962, December 2009. ISSN 1532-4435. URL `http://dl.acm.org/citation.cfm?id=1577069.1755883`. (Cited on page 88.)

[200] J. Herlocker, J. Konstan, L. Terveen, and J. Riedl. Evaluating collaborative filtering recommender systems. *ACM Trans. Inf. Syst.*, 22(1):5–53, January 2004. ISSN 1046-8188. doi: 10.1145/963770.963772. URL `http://doi.acm.org/10.1145/963770.963772`. (Cited on page 88.)

[201] G. Schröder, M. Thiele, and W. Lehner. Setting Goals and Choosing Metrics for Recommender System Evaluations. In *UCERSTI: Proceedings of the 2nd Workshop on User-Centric Evaluation of Recommender Systems and Their Interfaces*, Chicago, Illinois, 2011. (Cited on pages 89, 90, and 91.)

[202] R. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval*. Addison-Wesley Publishing Company, USA, 2nd edition, 2008. (Cited on pages 91 and 161.)

[203] Z. Gantner, S. Rendle, C. Freudenthaler, and L. Schmidt-Thieme. MyMediaLite: A free recommender system library. In *Proceedings of the 5th ACM conference on Recommender systems*, pages 305–308, New York, NY, 2011. (Cited on pages 91 and 161.)

[204] T. Bogers and A. van den Bosch. Recommending scientific articles using citeulike. In *RecSys '08: Proceedings of the 2th ACM Conference on Recommender Systems*, pages 287–290, Lousanne, Switzerland, 2008. (Cited on pages 91 and 161.)

[205] B. Arzanian, F. Akhlaghian, and P. Moradi. A Multi-Agent Based Personalized Meta-Search Engine Using Automatic Fuzzy Concept Networks. In *WKDD '10: Proceedings of the 3rd International Conference on Knowledge Discovery and Data Mining*, pages 208–211, Phuket, Thailand, 2010. (Cited on page 91.)

[206] J. Hebeler, M. Fisher, R. Blace, and A. Perez-Lopez. *Semantic Web Programming*. Wiley Publishing, 2009. ISBN 047041801X, 9780470418017. (Cited on pages 107, 108, and 148.)

[207] J. Allen and A. Frisch. What's in a Semantic Network? In *ACL '82: Proceedings of the 20th Annual Meeting on Association for Computational Linguistics*, pages 19–27, Toronto, Canada, 1982. (Cited on page 122.)

[208] J. Ge and Y. Qiu. Concept Similarity Matching Based on Semantic Distance. In *SKG '08: Proceedings of the 4th international conference on Semantics, Knowledge and Grid*, pages 380–383, Beijing, China, 2008. (Cited on pages 125 and 191.)

[209] U. Lösch, S. Rudolph, D. Vrandečić, and R. Studer. Tempus fugit. In *The Semantic Web: Research and Applications*, pages 278–292. Springer, 2009. (Cited on page 127.)

[210] J. Sangers, F. Hogenboom, and F. Frasincar. Event-driven ontology updating. In *Web Information Systems Engineering-WISE 2012*, pages 44–57. Springer, 2012. (Cited on pages 127 and 224.)

[211] J. Gosling. *The Java language specification*. Addison-Wesley Professional, 2000. (Cited on page 134.)

[212] J. Jena and P. Fuseki. Semantic web frameworks, 2004. (Cited on pages 134 and 146.)

[213] V. Benjamins, D. Fensel, S. Decker, and A. Perez. 2: building ontologies for the internet: a mid-term report. *International Journal of Human-Computer Studies*, 51(3):687–712, 1999. (Cited on page 135.)

[214] C. Liu, R. White, and S. Dumais. Understanding Web Browsing Behaviors Through Weibull Analysis of Dwell Time. In *SIGIR '10: Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 379–386, Geneva, Switzerland, 2010. (Cited on page 141.)

[215] E. Dijkstra. A note on two problems in connexion with graphs. *Numerische mathematik*, 1(1):269–271, 1959. (Cited on page 149.)

[216] H. Stuckenschmidt and A. Schlicht. Structure-based partitioning of large ontologies. In *Modular Ontologies*, pages 187–210. Springer, 2009. (Cited on pages 149 and 191.)

[217] P. Borlund. The iir evaluation model: a framework for evaluation of interactive information retrieval systems. *Information research*, 8(3), 2003. (Cited on page 169.)

[218] S. Dooms, T. De Pessemier, and L. Martens. A user-centric evaluation of recommender algorithms for an event recommendation system. In *RecSys '11: Proceedings of the 5th ACM Conference on Recommender Systems*, pages 67–73, Chicago, Illinois, 2011. (Cited on pages 169 and 193.)

[219] G. Kanji. *100 statistical tests*. SAGE Publication, 1993. ISBN 0803987056. (Cited on pages 172, 173, 174, 197, and 198.)

[220] H. Kondylakis, L. Koumakis, E. Kazantzaki, M. Chatzimina, M. Psaraki, K. Marias, and M. Tsiknakis. Patient empowerment through personal medical recommendations. *Studies in health technology and informatics*, 216: 11–17, 2015. (Cited on page 281.)

[221] V. Ostuni, T. Di Noia, R. Mirizzi, and E. Di Sciascio. *A Linked Data Recommender System Using a Neighborhood-Based Graph Kernel*. Springer International Publishing, 2014. (Cited on page 282.)

[222] D. Bogdanov, M. Haro, F. Fuhrmann, A. Xambó, E. Gómez, and P. Herrera. Semantic audio content-based music recommendation and visualization

based on user preference examples. *Information Processing & Management*, 49(1):13 – 33, 2013. (Cited on page 283.)

[223] I. Paraschiv, M. Dascalu, P. Dessus, S. Trausan-Matu, and D. McNamara. *A Paper Recommendation System with ReaderBench: The Graphical Visualization of Semantically Related Papers and Concepts*, pages 445–451. Springer, 2016. (Cited on page 283.)

[224] I. Paraschiv, M. Dascalu, S. Trausan-Matu, and P. Dessus. Analyzing the semantic relatedness of paper abstracts: An application to the educational research field. In *2015 20th International Conference on Control Systems and Computer Science*, pages 759–764, Bucharest, romania, 2015. (Cited on page 283.)

[225] T. Achakulvisut, D. Acuna, T. Ruangrong, and K. Kording. Science concierge: A fast content-based recommendation system for scientific publications. *arXiv preprint arXiv:1604.01070*, 2, 2016. (Cited on page 284.)

[226] J. Cordeiro, B. Antunes, and P. Gomes. Context-based recommendation to support problem solving in software development. In *2012 Third International Workshop on Recommendation Systems for Software Engineering (RSSE)*, pages 85–89, Zurich, Switzerland, 2012. (Cited on page 286.)

[227] M. Pozo, R. Chiky, and Z. Kazi-Aoul. Enhancing collaborative filtering using semantic relations in data. In *Computational Collective Intelligence. Technologies and Applications: 6th International Conference, ICCCI 2014, Seoul, Korea, September 24-26, 2014. Proceedings*. Springer International Publishing, 2014. (Cited on page 287.)

[228] S. Cadegnani, F. Guerra, S. Ilarri, M. Carmen Rodríguez-Hernández, R. Trillo-Lado, and Y. Velegrakis. *Recommending Web Pages Using Item-Based Collaborative Filtering Approaches*, pages 17–29. Springer International Publishing, Portugal, Coimbra, 2015. (Cited on page 288.)

[229] D. Ceccarelli, S. Gordea, C. Lucchese, F. Nardini, and R. Perego. When entities meet query recommender systems: Semantic search shortcuts. In *Pro-*

ceedings of the 28th Annual ACM Symposium on Applied Computing, SAC '13, pages 933–938, Coimbra, Portugal, 2013. ACM. (Cited on page 289.)

[230] O. Lee, M. Hong, J. Jung, J. Shin, and P. Kim. Adaptive collaborative filtering based on scalable clustering for big recommender systems. *Acta Polytechnica Hungarica*, 13(2), 2016. (Cited on page 290.)

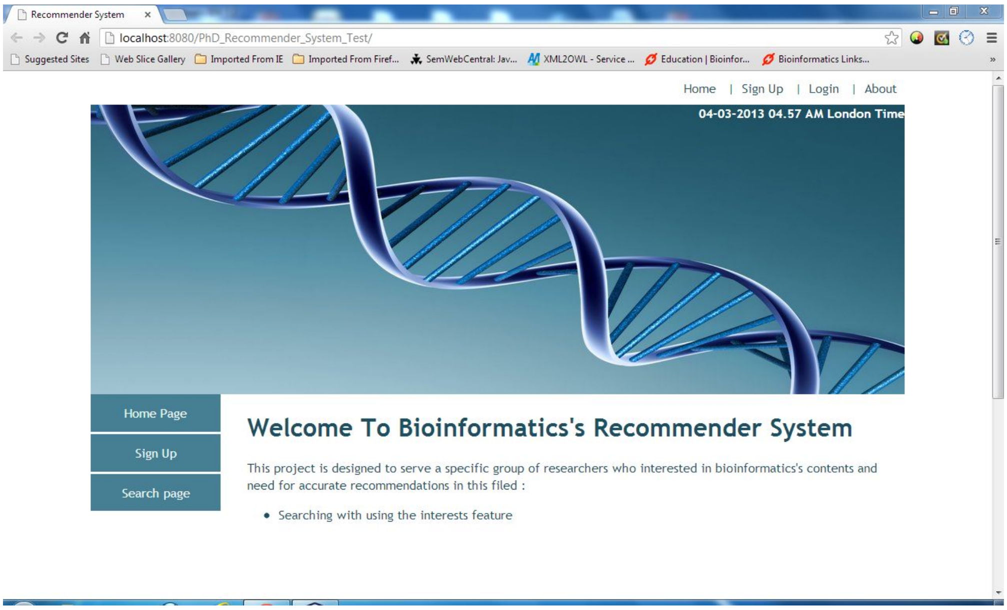
[231] M. Jiang, D. Song, L. Liao, and F Zhu. A bayesian recommender model for user rating and review profiling. *Tsinghua Science and Technology*, 20(6): 634–643, 2015. (Cited on page 292.)

[232] M. Moreno, S. Segrera, V. López, M. Muñoz, and Á. Sánchez. Web mining based framework for solving usual problems in recommender systems. a case study for movies? recommendation. *Neurocomputing*, 176:72–80, 2016. (Cited on page 293.)

[233] J. Lee, H. Kim, and S. Lee. A probability-based unified framework for semantic search and recommendation. *Journal of Information Science*, 39 (5):608–628, 2013. (Cited on page 294.)

Appdx A

A Searching Process Algorithm Snapshots of Implementations

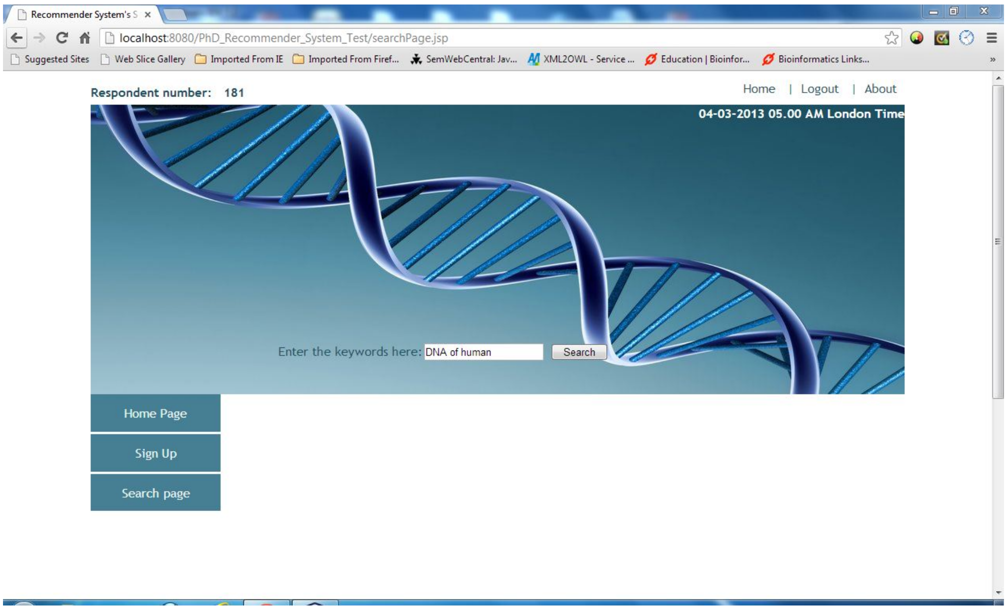


Welcome To Bioinformatics's Recommender System

This project is designed to serve a specific group of researchers who interested in bioinformatics's contents and need for accurate recommendations in this filed :

- Searching with using the interests feature

Figure 1: Search Engine Algorithm Interface



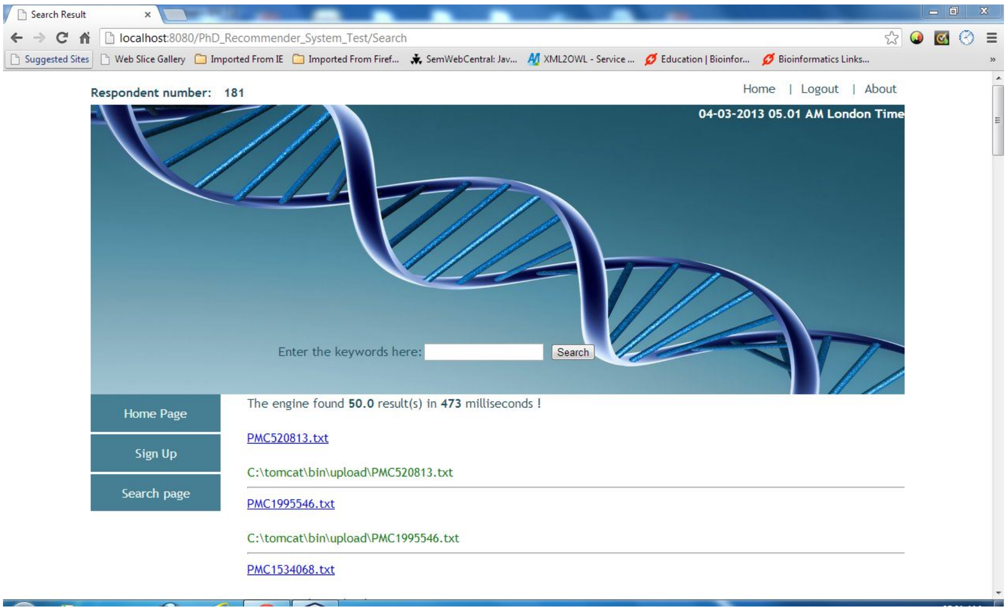
localhost:8080/PhD_Recommender_System_Test/searchPage.jsp

Respondent number: 181

04-03-2013 05.00 AM London Time

Enter the keywords here: DNA of human

Figure 2: User's Search Keyword



localhost:8080/PhD_Recommender_System_Test/Search

Respondent number: 181

04-03-2013 05.01 AM London Time

Enter the keywords here:

The engine found **50.0** result(s) in **473** milliseconds !

PMC520813.txt

C:\tomcat\bin\upload\PMC520813.txt

PMC1995546.txt

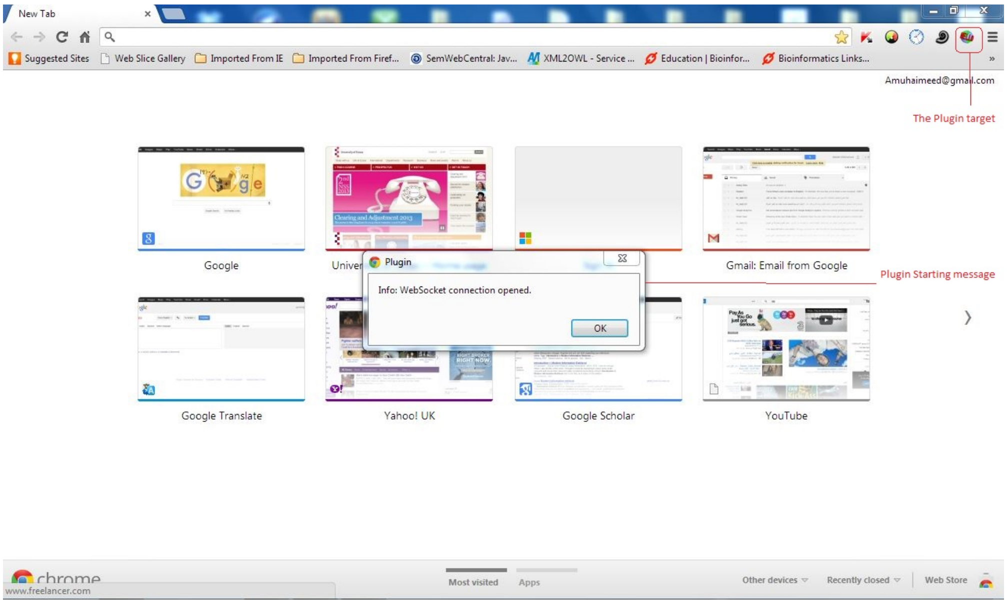
C:\tomcat\bin\upload\PMC1995546.txt

PMC1534068.txt

Figure 3: Top 100 Retrieved Documents of Search Process Algorithm

Appdx B

B Plugin's Snapshot of Implementations and Database Tables



Plugin

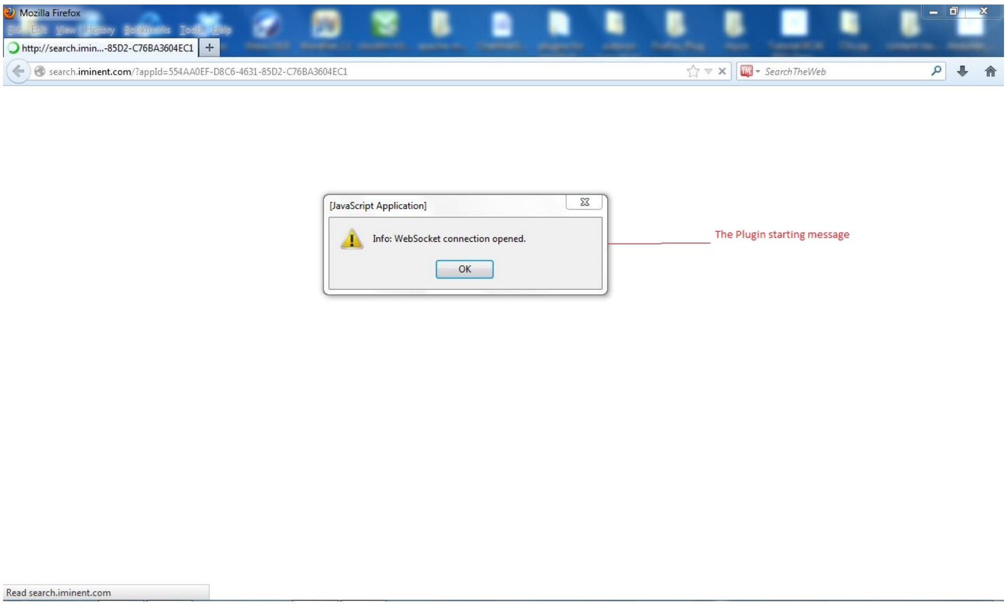
Info: WebSocket connection opened.

OK

The Plugin target

Plugin Starting message

Figure 4: Chrome Browser's Plug-in.


Mozilla Firefox
http://search.imin...-85D2-C76BA3604EC1
search.iminent.com/?appId=554AA0EF-D8C6-4631-85D2-C76BA3604EC1
SearchTheWeb
Read search.iminent.com

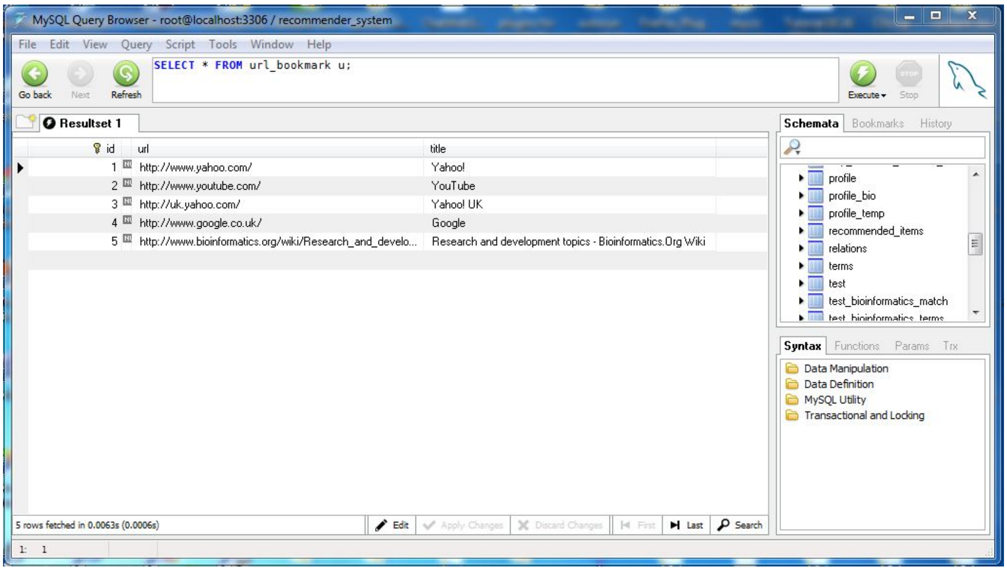
[JavaScript Application]

Info: WebSocket connection opened.

OK

The Plugin starting message

Figure 5: Mozilla Firefox Browser's Plug-in.



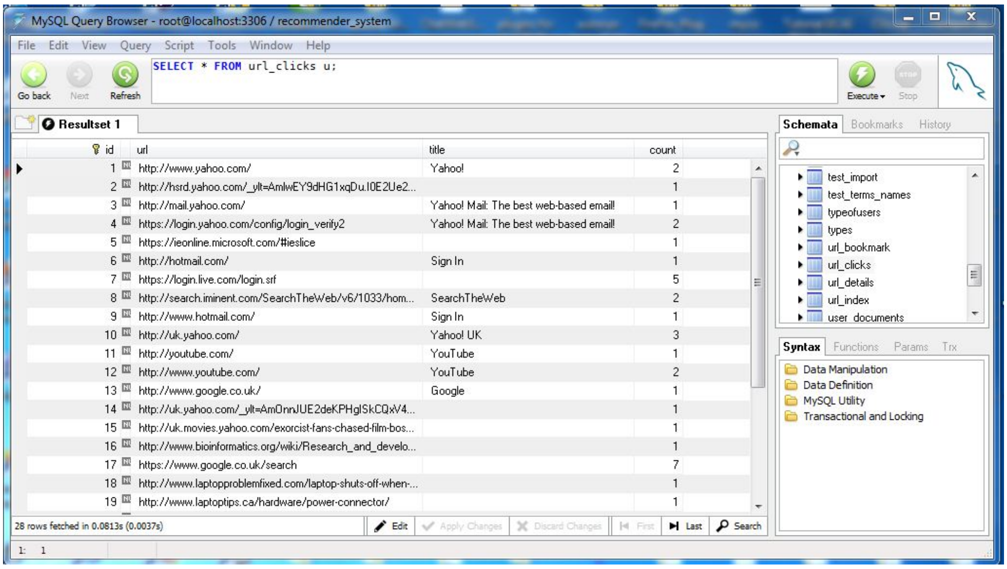
```
SELECT * FROM url_bookmark u;
```

id	url	title
1	http://www.yahoo.com/	Yahoo!
2	http://www.youtube.com/	YouTube
3	http://uk.yahoo.com/	Yahoo! UK
4	http://www.google.co.uk/	Google
5	http://www.bioinformatics.org/wiki/Research_and_develo...	Research and development topics - Bioinformatics.Org Wiki

5 rows fetched in 0.0063s (0.0006s)

1: 1

Figure 6: Bookmarked Website Database Table.



```
SELECT * FROM url_clicks u;
```

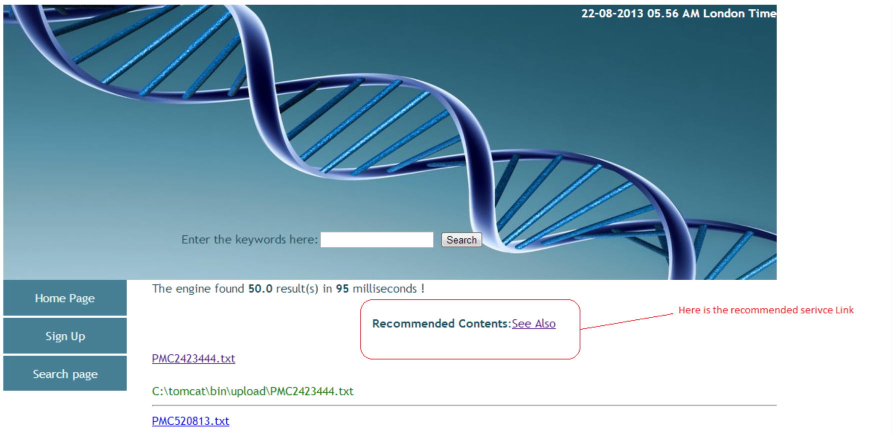
id	url	title	count
1	http://www.yahoo.com/	Yahoo!	2
2	http://hsrd.yahoo.com/_ylt=AmlwEY9dHG1xqDu.I0E2Ue2...		1
3	http://mail.yahoo.com/	Yahoo! Mail: The best web-based email!	1
4	https://login.yahoo.com/config/login_verify2	Yahoo! Mail: The best web-based email!	2
5	https://ieonline.microsoft.com/#ieslice		1
6	http://hotmail.com/	Sign In	1
7	https://login.live.com/login.srf		5
8	http://search.iminent.com/SearchTheWeb/v6/1033/hom...	SearchTheWeb	2
9	http://www.hotmail.com/	Sign In	1
10	http://uk.yahoo.com/	Yahoo! UK	3
11	http://youtube.com/	YouTube	1
12	http://www.youtube.com/	YouTube	2
13	http://www.google.co.uk/	Google	1
14	http://uk.yahoo.com/_ylt=AmOnnJUE2deKPHglSkCQxV4...		1
15	http://uk.movies.yahoo.com/exorcist-fans-chased-film-bos...		1
16	http://www.bioinformatics.org/wiki/Research_and_develo...		1
17	https://www.google.co.uk/search		7
18	http://www.laptopproblemfixed.com/laptop-shuts-off-when-...		1
19	http://www.laptoptips.ca/hardware/power-connector/		1

28 rows fetched in 0.0813s (0.0037s)

Figure 7: Clicked Links Database Table.

Appdx C

C Recommendation Service Interface



The engine found **50.0** result(s) in **95** milliseconds !

Recommended Contents: See Also

Here is the recommended serivce Link

PMC2423444.txt

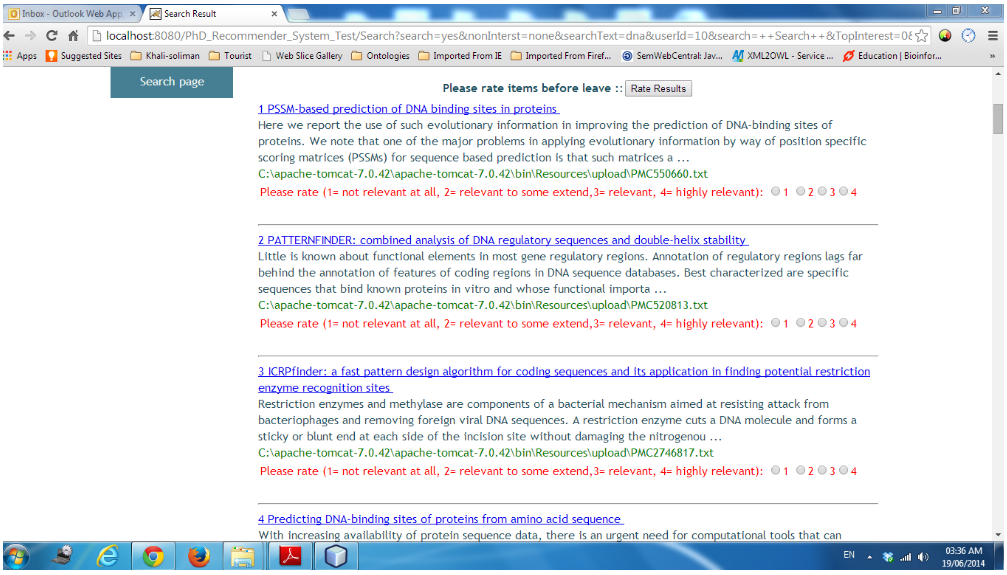
C:\tomcat\bin\upload\PMC2423444.txt

PMC520813.txt

Figure 8: Recommendation Service Interface

Appdx D

D Results rating Interface



Please rate items before leave :: Rate Results

1 PSSM-based prediction of DNA binding sites in proteins

Here we report the use of such evolutionary information in improving the prediction of DNA-binding sites of proteins. We note that one of the major problems in applying evolutionary information by way of position specific scoring matrices (PSSMs) for sequence based prediction is that such matrices a ...

C:\apache-tomcat-7.0.42\apache-tomcat-7.0.42\bin\Resources\upload\PMC550660.txt

Please rate (1= not relevant at all, 2= relevant to some extend,3= relevant, 4= highly relevant): ○ 1 ○ 2 ○ 3 ○ 4

2 PATTERNFINDER: combined analysis of DNA regulatory sequences and double-helix stability

Little is known about functional elements in most gene regulatory regions. Annotation of regulatory regions lags far behind the annotation of features of coding regions in DNA sequence databases. Best characterized are specific sequences that bind known proteins in vitro and whose functional importa ...

C:\apache-tomcat-7.0.42\apache-tomcat-7.0.42\bin\Resources\upload\PMC520813.txt

Please rate (1= not relevant at all, 2= relevant to some extend,3= relevant, 4= highly relevant): ○ 1 ○ 2 ○ 3 ○ 4

3 ICRPfinder: a fast pattern design algorithm for coding sequences and its application in finding potential restriction enzyme recognition sites

Restriction enzymes and methylase are components of a bacterial mechanism aimed at resisting attack from bacteriophages and removing foreign viral DNA sequences. A restriction enzyme cuts a DNA molecule and forms a sticky or blunt end at each side of the incision site without damaging the nitrogenou ...

C:\apache-tomcat-7.0.42\apache-tomcat-7.0.42\bin\Resources\upload\PMC2746817.txt

Please rate (1= not relevant at all, 2= relevant to some extend,3= relevant, 4= highly relevant): ○ 1 ○ 2 ○ 3 ○ 4

4 Predicting DNA-binding sites of proteins from amino acid sequence

With increasing availability of protein sequence data, there is an urgent need for computational tools that can

Figure 9: Results Rating Interface.

Appdx E

E Questionnaire and Tasks for Assessing Our Recommender Approach

This section contains questionnaires that will be distributed to five groups of participants where one of these groups will use our approach and the other questionnaires will be filled out by the other groups who are going to use the other approaches that have been considered for the evaluation process. The following sections will contain the questionnaires.

E.1 Questionnaire for Assessing Our Recommender Approach

Dear participant, Thank you for participating in our study, your time, effort and feedback are very much appreciated. The purpose of this questionnaire is to extract information on the views of the participants of our experiment on the recommender system that they used. The questionnaire should be filled by each participant after he/she has using the recommender system. Please note that the results of this study will be presented by aggregating and anonymising the participants' responses.
Please answer the following questions after you have finished performing the tasks/queries that you have submitted to the recommender system.

- **The items recommended to me matched my interests.**
  (Please choose from 1-5, where 5: Strongly Agree, 4: Agree, 3: Neither Agree nor Disagree, 2: Disagree, 1: Strongly Disagree)

  1 2 3 4 5

- **The recommendations provided to me were useful.**
  (Please choose from 1-5, where 5: Strongly Agree, 4: Agree, 3: Neither Agree nor Disagree, 2: Disagree, 1: Strongly Disagree)

  1 2 3 4 5

- **Overall, I have been satisfied with the services provided by the recommender system.**
  (Please choose from 1-5, where 5: Strongly Agree, 4: Agree, 3: Neither Agree nor Disagree, 2: Disagree, 1: Strongly Disagree)

  1 2 3 4 5

- **The items recommended to me are diverse.**
  (Please choose from 1-5, where 5: Strongly Agree, 4: Agree, 3: Neither Agree nor Disagree, 2: Disagree, 1: Strongly Disagree)

  1 2 3 4 5

- **The recommendation I have received better fits my interests than what I may receive from other search or recommender systems that I have used in the past.**
  (Please choose from 1-5, where 5: Strongly Agree, 4: Agree, 3: Neither Agree nor Disagree, 2: Disagree, 1: Strongly Disagree)

1 2 3 4 5

- **The recommended items cover a broad range of specific interests that I have in this area.**
  (Please choose from 1-5, where 5: Strongly Agree, 4: Agree, 3: Neither Agree nor Disagree, 2: Disagree, 1: Strongly Disagree)

1 2 3 4 5

- **The recommender system helps me to discover new articles.**
  (Please choose from 1-5, where 5: Strongly Agree, 4: Agree, 3: Neither Agree nor Disagree, 2: Disagree, 1: Strongly Disagree)

1 2 3 4 5

- **If a recommender such as this exists, I will use it to find articles to read.**
  (Please choose from 1-5, where 5: Strongly Agree, 4: Agree, 3: Neither Agree nor Disagree, 2: Disagree, 1: Strongly Disagree)

1 2 3 4 5

Thank you for your time and effort in participating in this study.

E.2 Tasks for examining different criteria in our recommender approach

- Task 1:
  - Job: Search about general concepts that explain bioinformatics field.

  - Description: Suppose you want to read general topics that explain bioinformatics such as definitions, advantages, drawbacks, history..etc.

- Task 2:
  - Job: Search for articles that discuss tools which are useful in bioinformatics.
  - Description: Suppose you are bioinformatician and you are looking for a tool that can help you to do a specific task in bioinformatics such as alignment, annotation etc.

- Task 3:
  - Job: Search for articles that discuss courses which explain programming languages that can be used for bioinformatics.
  - Description: Suppose you are bioinformatician and you are looking for a tool that can help you to program something in bioinformatics such as Matlab, Perl, etc.

- Task 4:
  - Job: Search for articles that discuss protein, gene, and sequence ontologies.
  - Description: Suppose you are bioinformatician and you are searching for articles which mention some bioinformatics ontologies such as PO, GO, SO etc.

- Task 5:
  - Job: Search for article that mentioned some bioinformatics journals.
  - Description: Suppose you are bioinformatician and you are searching for articles that mention academic journals such as BMC Bioinformatics, etc.

Appdx F

F Questionnaire and Tasks for Assessing Semantic Similarity Method

This section has questionnaire and tasks that have been passed over participants in order to examine different functionalities that provided in our approach. Every participant in each group will perform eight tasks during the experiment and answering the exist questionnaire whenever he finish the experiment.

F.1 Questionnaire for Assessing Semantic Similarity Method

Dear participant, Thank you for participating in our study, your time, effort and feedback are very much appreciated. The purpose of this questionnaire is to extract information on the views of the participants of our experiment on the recommender system that they have used. The questionnaire should be filled by each participant after he/she has used the recommender system. Please note that the results of this study will be presented by first anonymising the participants' responses and then aggregating them.
Please answer the following questions after you have finished performing the tasks/queries that you have submitted to the recommender system.

- **The items recommended to me matched my interests.**
  (Please choose from 1-5, where 5: Strongly Agree, 4: Agree, 3: Neither Agree

nor Disagree, 2: Disagree, 1: Strongly Disagree)

1 2 3 4 5

- **The recommendations provided to me were useful.**
  (Please choose from 1-5, where 5: Strongly Agree, 4: Agree, 3: Neither Agree nor Disagree, 2: Disagree, 1: Strongly Disagree)

  1 2 3 4 5

- **The recommendations suggested by the system uncovered articles or information that was new to me but very useful.**
  (Please choose from 1-5, where 5: Strongly Agree, 4: Agree, 3: Neither Agree nor Disagree, 2: Disagree, 1: Strongly Disagree)

  1 2 3 4 5

- **Overall, I have been satisfied with the services provided by the recommender system.**
  (Please choose from 1-5, where 5: Strongly Agree, 4: Agree, 3: Neither Agree nor Disagree, 2: Disagree, 1: Strongly Disagree)

  1 2 3 4 5

- **The items recommended to me are diverse.**
  (Please choose from 1-5, where 5: Strongly Agree, 4: Agree, 3: Neither Agree nor Disagree, 2: Disagree, 1: Strongly Disagree)

  1 2 3 4 5

- **The recommendation I have received better fits my interests than what I may receive from other search or recommender systems that I have used in the past.**
  (Please choose from 1-5, where 5: Strongly Agree, 4: Agree, 3: Neither Agree nor Disagree, 2: Disagree, 1: Strongly Disagree)

  1 2 3 4 5

- **The recommended items cover a specific interest and its associations that I have in this area.**
  (Please choose from 1-5, where 5: Strongly Agree, 4: Agree, 3: Neither Agree nor Disagree, 2: Disagree, 1: Strongly Disagree)

  1 2 3 4 5

- **The recommended items stem from or are associated in some way with your initial query and your preferences.**
  (Please choose from 1-5, where 5: Strongly Agree, 4: Agree, 3: Neither Agree nor Disagree, 2: Disagree, 1: Strongly Disagree)

  1 2 3 4 5

- **The recommender system helps me to discover new articles.**
  (Please choose from 1-5, where 5: Strongly Agree, 4: Agree, 3: Neither Agree nor Disagree, 2: Disagree, 1: Strongly Disagree)

  1 2 3 4 5

- **If a recommender such as this exists, I will use it to find articles to read.**
  (Please choose from 1-5, where 5: Strongly Agree, 4: Agree, 3: Neither Agree nor Disagree, 2: Disagree, 1: Strongly Disagree)

1 2 3 4 5

Thank you for your time and effort in participating in this study.

F.2 Tasks for Assessing Semantic Similarity Method

- Task 1:
  - Job: Search about general concepts that explain the bioinformatics field.
  - Description: Suppose you want to read general topics that explain bioinformatics such as definitions, advantages, drawbacks, history, etc.
- Task 2:
  - Job: Search for articles that discuss tools which are useful in bioinformatics.
  - Description: Suppose you are bioinformatician and you are looking for a tool that can help you to do a specific task in bioinformatics such as alignment, annotation etc.
- Task 3:
  - Job: Search for articles that discuss courses which explain programming languages that can be used for bioinformatics.
  - Description: Suppose you are bioinformatician and you are looking for a tool that can help you to program something in bioinformatics such as Matlab, Perl, etc.
- Task 4:
  - Job: Search for articles that discuss protein, gene, and sequence ontologies.
  - Description: Suppose you are bioinformatician and you are searching for articles which mention bioinformatics ontologies such as the protein

ontology (PO), gene ontology (GO), the sequence ontology (SO) or any other ontology that you may be aware of.

- Task 5:
  - Job: Select one of your own interests in the field of bioinformatics, then search for articles that are relevant to this interest you have selected.
  - Description: suppose you are interested in specific concept such as DNA (as an example) and it has been matched to tools used for extraction DNA in our system, so select DNA from your preferences and assess whether retrieved articles are about DNA and tools used for extraction DNA and their associations.
- Task 6:
  - Job: Log in to the system and press on the "See Recommendations" button (which will retrieve recommendations based on previous preferences that you have entered when registered in the system).
  - Description: Suppose you are interested to get recommendations on the preferences that you already have entered when you created your user profile in the recommender system. Examine the recommendations provided and assess if they match your interests.
- Task 7:
  - Job: Search for articles that discuss a specific gene or protein.
  - Description: Suppose you want to get recommendations on articles for a specific gene or protein. Once the results are retrieved, evaluate them based on how well the returned results match your initial query (as submitted) and your preferences.
- Task 8:
  - Job: This task complements the previous one, so please repeat the same phrase that you have entered to search for articles that discuss a specific gene or protein.

- Description: Suppose you want to get recommendations on articles for a specific gene or protein. Once the results are retrieved, evaluate them based on whether the retrieved results include articles that stem from or are associated in some way with your initial query and your preferences.

Appdx G

G Comparison between Different Recommender Systems

Table G.1: Comparison between Different Recommender Systems

		Features				
Types	Recommender Systems	Users Profile	Semantics	Resource(s)	Exploiting Search	Shortcomings
Content Based Filtering	Mirizzi et al.[159] proposed a recommender approach to provide recommendations on movies that exploits LOD dataset, which represents an ontology for multiple resources. It is designed as a plug-in that connects to Facebook in order to provide recommendations.	Yes, but not fully automatic	Yes, but limited	Multiple	No	Does not exploit semantic relations and associations and employs them successfully for enhancing recommendations. Also, it does not have an ontological user profile.

	Kondylakis et al.[220], provided a framework that integrates the patient's search engine with automatic personalised recommendations for individual patients based on their preferences, which are stored in their profiles and in their medical case file.	Yes, but not ontological nor fully dynamic	Yes, but it is very limited	Multiple	Yes	Although this work provides exploitation of the user's search for recommendations, this work does not take users' queries instantaneously and provides recommendations based on their submitted query as well as their preferences, which are stored in the users' profiles. It does this periodically by considering user queries as preferences. Moreover, the database rules, which are used for reasoning, are making the recommendations too general by refining the user query to match their defined category for a reasoning purpose. Therefore, suppose the patient is looking for information about flu, with the current rule used in this approach, this keyword will be generalised to match the word "Disease" and will then be reasoned over different resources as "Disease", and this may lead to inaccurate recommendations for what the user is looking for. Finally, this approach does not use a dynamic ontological user profile that will support the user to discover new facts related to his preferences and to receive up-to-date recommendations.

	Ostuni et al. [221] illustrated a content-based recommender approach that uses an LOD dataset and takes advantage of *a neighbourhood-based graph kernel*. The kernel method is able to compute the similarity between items by matching their local neighbourhood graph. This approach has used the Movielens dataset in order to evaluate their recommender approach.	Yes, but not Fully automatic and not ontological	Yes, but limited	Multiple	No	This work considers a single knowledge graph, i.e. DBpedia, to extract knowledge without applying its semantic extraction to other multiple resources that may be connected to DBpedia. Moreover, this work does not exploit semantics included in LOD successfully and does not have a fully automated ontological user profile.
	Kim et al. [177], provided a content recommender system that uses personal health records (PHR) and user-submitted queries about specific problems that he/she has had before. Moreover, this approach exploits ontologies to provide recommendations on ailments that are relevant to a user's history and his/her submitted query.	Yes, but not a dynamic user profile	Yes	Yes	Yes	This approach does not apply SPARQL when it extracts data from ontologies for a reasoning purpose. The way it is used here is to extract all data included in ontologies for reasoning and this will consume memory and will not able to handle huge data. Moreover, this approach does not have a dynamic user profile; thus, users will not receive up-to-date and accurate recommendations.

	Bogdanov et al. [222] illustrated a content-based recommendations method that produces semantic representation for user's preferences based on the audio that he/she preferred to listen to.	Yes, but not automated and not ontological	Yes, but limited	Single	No	The problem with this method occurs when similar items do not have any relation or connection with the selected tracks or items. This can show the importance of the existence of an inference method that is able to find any relationship or similarity between items stored in the user profile and recommended items. Also, there is a need for a method that is able to update a user preference automatically to avoid recommending irrelevant items to the user.
	Paraschiv et al. [223] provided a paper recommender system that represents an extension of the work discussed in [224], in term of the semantic view. It allows users to submit queries in natural language text and then recommends the most relevant papers. Moreover, it produces coherent concepts that relate to the submitted query to find the relevant keywords from documents that have a semantic relation to the submitted query.	Very limited, not ontological or automatic	Yes, but limited	Single	Yes	This approach tries to enhance recommendations by using query expansion. However, it uses a single resource of data to provide recommendations and uses a very limited user profile that is not ontological or automated. Moreover, it only tries to exploit the semantic similarity between texts without employing any relation or introducing any inference method that is able to discover any new information.

	Achakulvisut et al. [225] illustrated a recommender system for relevant publications that allows scientists to find related articles from a wide range of scholars throughout the world. Thus, it introduces an algorithm supported by Python library to perform recommendations from a set of publications that are similar to those the researcher has read before. Moreover, this method is intended to provide new articles and provides near real-time recommendations to researchers.	Very limited, not ontological or automatic	Yes, but limited	Single	No	This approach calculates similarity between similar contents; however, it does not apply any inference method to perform further reasoning over the processed scientific papers in the used dataset. Also, this work does not use several resources to exploit the overlapped information between these resources in order to discover new facts that may lead to enhanced recommendations. Moreover, it uses a rudimentary user profile that is not ontological and not equipped with a method that is responsible for adaptation of preferences.

	Google API[1,2], an API that can be set up to provide a search engine and recommendations that help users to find their needs in specific datasets.	Yes, but it is not ontological or adaptive.	No	Multiple	Yes	This API provides both search and recommendations services in a specific dataset. However, it suffers from limitations such as the fact that it does not consider semantics between concepts in single or multiple resources, and this may support both the search and the recommendations with extra facts and information that makes them more accurate. Moreover, it does not have an inference method that is able to reason between different resources that extract semantic relations and associations in order to leverage them in enhancing recommendations. Also, it does not support the ontological user profile, which may add extra information to the user profile and help to discover new facts that lead for more accurate recommendations.

[1] https://cloud.google.com/prediction/docs

[2] https://developers.google.com/custom-search/?hl=en

	Cordeiro et al. [226], suggested an approach that integrated into the programmer's work environment (as a plug-in in Eclipse) and allowed them access to questions/answers on Web resources and gave them recommendations based on the fail or exception they have. Since the exception in this approach represents the key search that developers need to receive recommendations about, the approach exploits this to provide recommendations that are designed to address this problem or to give possible solutions to eliminate this exception.	No	No	Multiple	Yes	Although this approach provides useful recommendations with regard to the unexpected exceptions that occur for the programmers during their programming, this approach lacks semantics that can help to enhance the accuracy and quality of recommendations. This exploitation can be done be adding a method that reasons through different Web resources and extracts sets of questions and answers that were asked before and then extracts any semantic relations or hidden associations that occur as a result of information overlapping between different resources and then exploits them to enhance recommendations. Also, it can help the developers to discover new information and facts that reduce the time and effort it takes to address the current problem or to avoid similar exceptions from occurring in the future. This approach does not have a user profile that can help their recommender approach to provide recommendations to each programmer while taking into account his/her useful recommendations (preferences) that he/she has rated. Also, this profile should be adaptable and ontological in order to receive up-to-date recommendations and to exploit the extra information gained from the exploited ontology in the user profile to enhance recommendations.

	Livne et al [178], illustrated an approach called *CiteSight*, which designed a personalised recommendation of citations to author groups of multiple assigned tasks. This approach allows the author to write keywords, and then it will provide relevant recommendations to the given keywords. This is called *inline* recommendations.	Yes, but not ontological	No	Multiple	Yes	This does not exploit semantics between different papers in order to help users receive more accurate recommendations that lead them to discover more citations or papers relevant to their interested topics. Moreover, this approach is not equipped with an ontological user profile that may contribute to provide extra knowledge that will lead users to discover new papers or citations that help them in their writing.
Collaborative Based Filtering	Pozo et al. [227] suggested a recommendation method that depends on semantic data to ensure high quality and accurate recommendations. Moreover, they have used a distributing collaborative filtering algorithm based on Alternating Least Squares (ALS) to ensure scalability in their recommender system.	Very basic user profile, not automated or ontological	To some extent	Single	No	This approach is lacking an inference method that can be applied over the domain ontology and can used to infer new information about different movies. Moreover, it uses a single ontology to provide recommendations. By considering multiple resources (i.e. ontologies), it can add more enhancement to their recommendations. Also, this approach suffers from the general problem that appears as a result of applying collaborative filtering such as sparsity and the cold start problem.

	Cadegnani et al. [228] proposed a recommender system that employs three methods to enhance the recommendations. All these methods intend to exploit webpage details, such as information included in their structures and their log files, to enhance the accuracy of the webpage recommendations. Moreover, the contexts of the users (i.e. their running sessions when they are browsing a particular website) are also taken into account to support the recommendations process in this approach.	Yes, but not ontological or automatic	Yes, but limited	Single	No	This approach tries to enhance recommendations by exploiting semantics between webpages and employs other factors such as the location of the browsed webpage and the structure of the website, etc. However, it is lacking multiple resources that can be exploited to find similar webpages and it is lacking an inference method that contributes to finding webpages that are more similar to the browsed one and which allows users to discover more information to enrich their knowledge. Moreover, it only considers the current webpage (the browsed page) to decide on the user preference when it does not give enough information to conclude about the user's preferences, and it lacks an automated method to keep the user profile updated.

	Ceccarelli et al. [229], suggested an approach to query recommendations enhancement, which is called Semantic Search Shortcuts (S^3). This approach enhances query recommendations by suggesting the top-k of the most successful queries (a query is classified as successful when the user accessed the results returned by this query, but is classed as unsuccessful if he/she did not access them) for the list of other users. This approach annotates the user's query with something called *virtual documents* (which is the title and content of each successful query) in order to map its relevant entities to exploit semantics that could be gained from these entities for better recommendations.	Yes, but not ontological or adaptable	Yes, but in a very limited way.	Single	Yes	This approach is not fully automated and this may effect the query annotation and cause misleading annotations. This is due to the frequent changes and updates in the Web documents. Therefore, during this time, the virtual document's contents may change and may lead to misleading annotations on specific queries, which makes semantic exploitations very limited and weakens the accuracy of the provided recommendations. Furthermore, this approach uses a very limited user profile that is only concerned with a query; however, there are other factors such as browsing behaviours that could be included, which lead to better recommendations since it helps to discover user preferences that contribute to recommend that user with accurate query recommendations based on his/her own preferences. Moreover, this approach does not exploit ontology to construct the user profile, which may add extra knowledge that helps to enhance the accuracy of the provided recommendations. Furthermore, this approach does not use an adaptive user profile, which may cause that user to receive inaccurate recommendations.

	Lee et al. [230] introduced a collaborative filtering approach that contributes to address the problem of scalability by constriction time involvement to formulate neighbourhood. Moreover, they addressed the data sparsity problem by leverage the feature of users and items as vectors that learn gradually. Moreover, their method was performed by four main elements, which are: *i) scalable clustering, ii) recommendations, iii) preference prediction, and iv) learning*. These four elements allow users' interactions to enrich the cluster model to produce better recommendations.	Yes, but not ontological or automatic	No	Single	No	This work provides a solution to common problems that occur in collaborative filtering, especially with the existence of big data such as sparsity, cold start, addressing the scalability performance, etc. However, it does not consider semantics that may exist between processed data in order to exploit them and enhance recommendations. Moreover, it does not consider multiple resources and employs an inference method that is able to discover facts and information to enhance recommendations. Also, it does not use any method that maintains user preferences in an up-to-date manner.

Hybrid Filtering	Meymandpour and Davis [164] provided a hybrid recommendation method for the Movielens dataset. It uses two recommendation filters to overcome the cold-start problem and to provide better movie recommendations to the users. These filters are collaborative and use semantic-analysis for LOD to provide better recommendations.	Yes, but not ontological or automatic	Yes, but it is very limited	Multiple	No	Even though this method enhanced the quality of the recommendations, it still suffers from limitations that restrict its performance at the level of accuracy in the provided recommendations. This method exploited some semantic relations included in LOD but in a limited way, as this dataset contains several types of relations (such as *has-part* or any other relations that could be discovered from information overlapping) that also can be exploited to enhance recommendations. Moreover, it does not have an automatic ontological user profile, whereby the ontological profile could help to infer new relations and information that could exist in the ontology combined with user preferences and the LOD dataset. This ontological user profile should work with an automatic method that is responsible for adding, deleting and updating user preferences that may change over time to keep recommended items updated.

	Jiang et al. [231] proposed a model for recommendations that integrates collaborative filtering and content-based filtering in order to enhance the accuracy of deciding user's preferences and to address or reduce some of the common problems in the recommender systems such as cold-start. Thus, their model is a Bayesian model, which is called the User Rating and Review Profile (URRP). It connects the User Rating Profile (URP) with Latent Dirichlet Allocation (LDA) fluently. Moreover, it has the ability to express the rating in a different way, which concluded by supporting the learning of user-rating behaviours with review topics, which distinguishes this model from the other relevant works.	Yes, but not an ontological user profile	No	Single	No	There are two shortcomings of this approach: it does not employ ontologies to represent the user profile, which can help to infer new information and relations that would not appear in the standard way. The second limitation of this approach is that it does not consider semantic information such as between user ratings and user profile review.

	Moreno et al.[232] illustrated a framework that contributes to addressing the most common problems or drawbacks of different existing recommender systems. These problems can be summed up as the following: *i) scalability, ii) cold-start, iii) first rate, and iv) sparsity.* Although these problems have been addressed before, they assume that nothing before has treated or overcome all these problems together. Moreover, they have used movie recommendations to assess the performance of their framework to address the aforementioned problems and they confirm that their framework is flexible enough to be applicable in any other domain.	Ontological user profile, but not automated	Very limited	Single	No	Although this approach provides a set of semantic Web techniques to overcome some problems in recommendations such as sparsity and cold start, etc., it does not include an inference method that is able to infer new semantic facts or information that exist in the considered ontology in order to provide new recommendations to the user. Moreover, it does not consider multiple resources (i.e. ontologies) that can be extracted from the inference method (in the case that they apply it in their work) in order to provide various recommendations that contain new data that may not appear in the single resource. Finally, this approach ignores the frequent changes that may occur in the user preference during that time, as it does not include any automatic method that is in charge of keeping user preferences updated.

	Lee et al. [233], provided a probabilistic framework that is used to enhance a semantic search and recommendations. This framework uses a hybrid search and collaborative filtering recommendations to overcome some of the popular problems that occur in most of the classical keyword search engines and in recommender systems such as semantic ambiguity, non-personalisation and sparsity. This framework tries to fulfil users' needs and documents by retrieving and recommending documents to him/her that have high semantic relevancy. This framework represents the semantic users' needs and documents with concepts from domain knowledge. Moreover, it uses a probabilistic model to represent the entities and their relationships, in which they are represented as a probabilistic graph, which consists of entities as random variables and in which their relationships are expressed as conditional probabilities.	Yes, but it is not adaptive and not fully automated	Yes	Multiple	Yes	The main drawback of this approach is the use of the probabilistic model to represent the entities and their relations. This is because this model is not quite sufficient for huge data that contain complicated relations. This model consumes a lot of time for calculating and representing the relationships between entities; it is also not quite accurate when it comes to complex relations. Thus, using this model may lead to a bad performance for this framework or may consume machine memory during the calculation process. Moreover, this framework is not supported with an adaptive and automatic user profile, which would ensure that users do not receive inaccurate recommendations.