UNIVERSITAT JAUME I

Departament d'Enginyeria i Ciència dels Computadors

Semantic-based approach for the discovery of Life Sciences web resources driven by rich user's requirements

Ph. D. dissertation

María Pérez Catalán

Supervised by Dr. Rafael Berlanga Llavori and Dr. Ismael Sanz Blasco

Castellón, September 2013

To Ricardo.
To my parents and brother.

Abstract

Web resources have been gaining popularity as providers of relevant data, whether those stored in datasets or those resulting from the execution of complex functions such as the alignment of protein sequences. Although the discovery of web resources has been largely studied, it is still a challenging research task due to the high dependency current search engines have on the characteristics of the available metadata. In some domains like Life Sciences, this dependency becomes even worse due to the heterogeneity of data.

Current web resource registries allow users to search for resources that fulfill their information needs. The discovery in these registries is mainly based on the use of well-defined metadata, which is usually limited and very specific, and on the string matching of the user's query keywords, which is hampered by the heterogeneity of data.

The main objective of this thesis is to assist the users in the discovery of the most appropriate resources for their information needs, specifically in the Life Sciences domain. The achievement of this objective implies addressing the main limitations of current web resource registries.

Firstly, web resource discovery is driven by the user's requirements and, therefore, the precision of its results depends on how well the user's information needs are described in the requirements specification. Thus, rich requirements specifications are assumed to obtain more precise results. In the proposed approach, the requirements specification consists of a rich description of both the functionality and relevant features of the required resource. Additionally, discovery parameters are customizable by the users in order to improve the accuracy of the process.

Secondly, the discovery depends heavily on the characteristics of the resources metadata. In many registries, resources are described with well-

defined metadata, e.g., categories, and with textual descriptions, which provide richer information but harder automatic processing. In order to alleviate this dependency, this thesis proposes a normalization process which addresses the heterogeneity of data, and automatically identifies relevant information implicitly described in the resources metadata. Then, the discovery of web resources considers the normalized data, reducing words mismatchings, alleviating the problem of using different vocabularies, and improving the characterization of resources.

Finally, whereas current registries provide the user with a list of resources without any information about their relevance to her requirements, in the proposed approach the user is prompted with a ranked list of resources according to the fulfillment of her information needs, and to the accomplishment of the user-defined features. In this way, the system assists the user until the end of the discovery process, providing her information relevant to the selection of the best suited resource.

The experimental evaluation performed on each phase of the discovery method demonstrates that the proposed techniques obtain good results. Moreover, the discovery method has been implemented as part of BioUSeR, an online tool for the discovery of Life Sciences web resources. In BioUSeR, the results of each phase of the discovery process are visualized, and the parameters and the data involved in the process are easily customized by the user. We have used BioUSeR to demonstrate the usefulness of our approach using real usage examples.

Keywords: Web resource discovery, requirements-driven methods, semantics, information retrieval.

Aportaciones, Conclusiones y Trabajo Futuro

Introducción

En los últimos años, la cantidad de datos publicados en Internet ha crecido a un ritmo vertiginoso. En la actualidad, instituciones, empresas y particulares están publicando información en Internet con el propósito de compartirla con otros usuarios. Mucha de esta información es accesible mediante recursos web, los cuales permiten, entre otras muchas cosas, recuperar información de bases de datos o ejecutar complejos algoritmos sobre la información dada. Estos recursos web están descritos con metadatos que suelen definir características importantes tanto de los recursos como de la información que contienen o procesan. La gran cantidad de recursos web disponibles actualmente, su heterogeneidad, y su distribución, hacen que encontrar el recurso adecuado para un determinado requisito se haya convertido, en muchos casos, en una ardua y reiterativa tarea.

Para facilitar la búsqueda, en Internet existen buscadores específicos y registros centralizados de recursos web, siendo estos últimos los más populares actualmente. Los registros de recursos web contienen los metadatos de los recursos incluídos y permiten al usuario realizar búsquedas basadas en dichos metadatos. En estos registros, los recursos suelen estar categorizados y descritos con un conjunto de etiquetas y una descripción textual. Los mecanismos de búsqueda más comunes son la navegación a través de taxonomías de categorías y la búsqueda mediante palabras clave. La búsqueda por categorías normalmente limita al usuario en la especificación de sus requisitos cuando éstos son muy específicos y no existen categorías con tal grado de especificación. Además de depender del grado de especificación

de la taxonomía, los resultados también dependen de la calidad de las categorizaciones de los recursos, ya que en algunos registros, por ejemplo BioCatalogue, muchos recursos no tienen categoría asignada. Por otro lado, la mayoría de los registros permiten también la búsqueda por palabras clave, que consiste en buscar aquellos recursos en los que aparece alguna de las palabras proporcionadas por el usuario. Esta búsqueda suele estar limitada por la variabilidad de las palabras y por la heterogeneidad de los vocabularios utilizados.

Por tanto, en los registros de recursos web actuales, la especificación de los requisitos de usuario suele ser poco representativa, lo que repercute directamente en la precisión de los resultados obtenidos.

Hay que añadir que al finalizar el proceso de búsqueda, la mayoría de los sistemas devuelven una lista de los recursos que han sido recuperados porque tienen alguna palabra en común con la consulta que el usuario ha proporcionado, o porque tienen asignadas las categorías seleccionadas por el usuario. Sin embargo, no le ofrecen al usuario ninguna información adicional acerca de la relevancia de cada recurso. Por tanto, podemos decir que los actuales sistemas de búsqueda no asisten al usuario en la búsqueda del mejor recurso según sus necesidades. Además, si el usuario no está satisfecho con los resultados obtenidos, tiene pocas posibilidades de personalizar su búsqueda, aparte de cambiar los requisitos iniciales.

Esta tesis está centrada en el dominio de las Ciencias de la Vida, en el cual la heterogeneidad de los datos es muy elevada debido a la falta de estándares aceptados por la comunidad. En este dominio, los investigadores publican, en forma de recursos web, los resultados de sus investigaciones y las aplicaciones utilizadas. En la actualidad, existen múltiples registros de recursos relacionados con las Ciencias de la Vida, por ejemplo, BioCatalogue y SSWAP. Sin embargo, la mayoría presentan las mismas limitaciones que los registros de ámbito general, agravadas aún más por la heterogeneidad de los datos.

Objetivos

El principal objetivo de esta tesis es diseñar un método de búsqueda de recursos web que guíe al usuario durante todo el proceso, permitiéndole personalizar la búsqueda y proporcionándole información relevante para la selección del recurso web más adecuado.

La búsqueda es completamente dirigida por los requisitos de usuario; por lo tanto, la especificación de éstos es clave para el éxito de la búsqueda. En esta tesis, los requisitos no sólo se refieren a la funcionalidad requerida, sino también a características de los recursos que son relevantes para el usuario. Además, para conseguir una buena especificación, el usuario debe poder modificar la información referente a sus requisitos en cada etapa del proceso, y también debe poder personalizar los parámetros de la búsqueda según sus necesidades.

Sin embargo, una especificación precisa de los requisitos de usuario no es suficiente para garantizar la obtención de los recursos adecuados, ya que ésta dependerá de las características de los metadatos de los recursos web. Así pues, tanto la especificación de los requisitos como el proceso de búsqueda tienen que ser totalmente independientes de cómo están descritos los recursos, es decir, de la estructura de sus metadatos y de los vocabularios utilizados.

Finalmente, y al hilo de asistir al usuario durante todo el proceso, éste debe recibir información relevante referente al proceso de búsqueda, así como información de los resultados obtenidos para facilitarle la selección del recurso más adecuado.

Metodología

Esta tesis está basada principalmente en el uso de técnicas de normalización semántica y técnicas de recuperación de información para la consecución de los objetivos anteriormente descritos.

Uno de los principales problemas de los motores de búsqueda de los registros actuales es la dependencia de las características de los metadatos de los re-

cursos. Cada vez más los recursos son descritos con descripciones textuales, lo que dificulta su procesamiento automático, ya que además de la heterogeneidad, presenta otros problemas como la ambigüedad o la descripción implícita de características relevantes en las descripciones textuales.

El método propuesto basa la búsqueda de recursos web en la normalización de todos los datos involucrados en el proceso de búsqueda, con el fin de no depender de las características de los metadatos de los recursos ni de las características de la especificación de los requisitos del usuario.

Esta tesis propone un método de normalización cuyo objetivo es describir y caracterizar los recursos web en un formato que el sistema pueda procesar de forma automática, y que alivie los problemas relacionados con la heterogeneidad y las características intrínsecas del lenguaje natural, como por ejemplo, la información descrita implícitamente. La normalización propuesta está basada en la anotación semántica de los datos y en la extracción automática de información relevante acerca del recurso web.

La búsqueda de recursos web se basa en la información normalizada tanto de los requisitos como de los recursos web. Esta búsqueda basada en semántica permite recuperar recursos descritos con diferentes vocabularios o descritos a diferente nivel de detalle, así como recursos relacionados. Además, gracias a la normalización, la búsqueda es independiente de la técnica utilizada para definir los requisitos del usuario, permitiendo de esta manera utilizar diferentes tipos de especificación según el tipo de usuario, con el fin de obtener especificaciones precisas de sus necesidades.

Por último, el modelo de recuperación propuesto en esta tesis genera una lista ordenada de los recursos recuperados según su relevancia respecto a los requisitos de usuario, basada en el cumplimiento tanto de la funcionalidad requerida como de las características definidas por el usuario.

Aportaciones

En primer lugar, la tesis realiza una revisión general de las diferentes arquitecturas y técnicas utilizadas para la búsqueda de recursos web, y luego se centra en el dominio de las Ciencias de la Vida, realizando una revisión

más exhaustiva de los registros de recursos web más populares en dicho dominio. A partir de esta revisión se extraen las principales limitaciones de los registros actuales, como son la baja representatividad de los requisitos de usuario, la falta de asistencia al usuario durante el proceso de búsqueda, y la alta dependencia de las características de la información involucrada.

La aportación principal de esta tesis es un método para la búsqueda de recursos web en el dominio de las Ciencias de la Vida, siendo el usuario la pieza fundamental durante todo el proceso. Este método consta principalmente de dos fases: *(i)* la normalización de la información y *(ii)* la búsqueda y ranking de recursos web.

La fase de normalización propuesta consta de dos partes: la anotación semántica y la extracción automática de información. La anotación semántica es realizada de forma automática por un anotador que es capaz de utilizar de forma simultánea múltiples recursos de conocimiento con el fin de ampliar la cobertura de las anotaciones y tratar problemas como la ambigüedad. Posteriormente, se aplican técnicas de extracción de conocimiento basadas en la semántica y en modelos probabilísticos que permiten identificar de forma automática características relevantes del recurso, mejorando así su caracterización.

Respecto a la búsqueda de los recursos web, la tesis propone un modelo de recuperación basado en semántica que tiene en cuenta la información normalizada, tanto de la especificación de los requisitos como de los recursos web. Además, el modelo de recuperación define una función de similitud utilizada para ordenar los recursos recuperados en función de su relevancia respecto a los requisitos, teniendo en cuenta tanto la funcionalidad como las características definidas por el usuario.

Todas las técnicas propuestas han sido evaluadas y comparadas con otras técnicas de recuperación de información. Además, para demostrar que el método propuesto mejora las limitaciones de los registros de recursos web actuales en el dominio de las Ciencias de la Vida, se ha comparado con BioCatalogue, uno de los registros más populares actualmente.

Por último, el método propuesto ha sido implementado como parte de BioUSeR, una aplicación online que permite la búsqueda de recursos web

bioinformáticos y muestra al usuario los resultados de cada fase, haciéndole así partícipe de todo el proceso.

Conclusiones y Trabajo Futuro

Esta tesis propone un método para la búsqueda de recursos web en el área de las Ciencias de la Vida cuyo objetivo es guiar al usuario durante todo el proceso de búsqueda.

El método propuesto está dirigido por los requisitos del usuario, que consisten en una descripción precisa de las necesidades de información del usuario. Esta descripción, que incluye tanto la funcionalidad requerida como las características relevantes, debe ser independiente de los formatos y vocabularios utilizados en los metadatos de los recursos, eliminando así cualquier limitación para el usuario.

Para aliviar la dependencia de las características de los datos, esta tesis propone un proceso de normalización, basado en la anotación semántica y en la extracción automática de información relevante, que reduce la heterogeneidad de los datos y permite obtener una caracterización precisa de los recursos de forma automática.

La búsqueda de recursos web se basa en la normalización de los datos y, por tanto, recupera recursos descritos con diferentes vocabularios y a diferentes niveles de detalle. Con el fin de asistir al usuario hasta el final, los recursos recuperados son ordenados según su relevancia respecto a los requisitos, la cual no se basa en el número de palabras en común, sino en el cumplimiento de la funcionalidad requerida por el usuario y la presencia de las características especificadas. Además, el método propuesto permite al usuario personalizar la búsqueda cambiando parámetros, como la relevancia de una determinada característica, o modificando la información extraída automáticamente acerca de sus requisitos.

Como trabajo futuro, existen múltiples aspectos de las técnicas propuestas que podrían ser mejoradas, así como futuras líneas de investigación surgidas a raíz del trabajo realizado en esta tesis.

Una de las técnicas propuestas que podría ser mejorada es la anotación semántica, concretamente su post-procesamiento, con el fin de obtener anotaciones más precisas. Por ejemplo, se podrían utilizar técnicas de disambiguación para reducir la ambigüedad de los textos, y técnicas de simplificación más precisas basadas en el contexto.

Otro aspecto importante que se podría mejorar es el ranking de los recursos. En la literatura existen múltiples técnicas que permiten mejorar el ranking de resultados como, por ejemplo, el feedback de usuario, los requisitos no funcionales, e incluso la consideración del contexto. En nuestra opinión, cuanta más información acerca del usuario se tenga en cuenta en el ranking, más preciso y más personalizado será.

Además de estas posibles mejoras de las técnicas ya implementadas, a raíz del trabajo realizado en esta tesis se han abierto nuevas líneas de investigación. Cabe destacar que aunque esta tesis esté centrada en el dominio de las Ciencias de la Vida, sus técnicas se pueden aplicar en cualquier otro dominio. Estas técnicas permiten recuperar recursos web, independientemente del tipo que sean y de cómo estén descritos.

Una posible línea de investigación podría ser la recuperación e integración de recursos de diferentes tipos, como imágenes, documentos o audio. Por ejemplo, nuestras técnicas podrían ser aplicadas en sistemas que almacenan diferentes tipos de recursos, como por ejemplo, sistemas de almacenamiento de recursos médicos que contienen imágenes, informes clínicos, etc. Toda la información almacenada, la información de los informes y los metadatos de las imágenes, podría ser normalizada por nuestro proceso de normalización, homogeneizando terminologías e incluso idiomas. A partir de estos datos normalizados, los usuarios podrían recuperar diferentes tipos de recursos para una misma consulta, o incluso recuperar recursos relacionados a través de las relaciones entre conceptos definidas en las ontologías. Además, con nuestro método también se podrían consultar recursos externos, como publicaciones científicas, con el fin de proporcionar al usuario información adicional.

Otra línea de investigación interesante es la composición de workflows a partir de los recursos recuperados por nuestro método. Actualmente, la com-

posición de workflows está muy limitada por la disponibilidad de metadatos bien definidos de los recursos web, principalmente información acerca de los parámetros de entrada y salida. En nuestra propuesta, esta limitación sería aliviada por los mecanismos de caracterización de los recursos web. Además, si se utiliza una técnica de especificación de requisitos que permita definir relaciones explícitas entre tareas, nuestro sistema podría ser utilizado para la composición de workflows.

Acknowledgements

During the last years, I have met very nice people to whom I have to be very grateful, because without them this thesis would not have been completed.

First of all, I would like to thank Rafael Berlanga, who admitted me in his research group (TKBG) and who has supervised my PhD. He has made many important contributions to this work, without which I would have felt lost a lot of times. I have learnt a lot with him, and I admire his ability to find alternatives and solutions to any problem. I would like also to express my gratitude to Ismael Sanz for introducing me in the research world. He has been the advisor of my final degree project, my Master's thesis and now, my PhD. I would also thank María José Aramburu because she has given me the chance to make my PhD, and because she has always given me valuable advice. You three have made possible that I am here today. Thanks!

Thanks to my colleagues in TKBG group. I know that without you this work would have been harder and frustrating. For me, it has been a pleasure to share these years with all of you.

I thank also my thesis committee members and reviewers for accepting to be part of this.

This thesis is also dedicated to my family, who has been the most important support for me during these years, without them I would not be here today. I would like to acknowledge specially my husband Ricardo for his extremely patience, for being always there, specially in the difficult days, and for his always encouraging words. Thanks!

Last but not least, I would dedicate this thesis to my friends, who did not usually understand what I was doing, but who have been always there.

This work was mainly supported by the predoctoral grant of Universitat Jaume I (PREDOC/2007/41), and by research projects from the Ministerio

de Economía y Competitividad (TIN2008-01825, TIN2011-24147) and the Fundació Caixa Castelló projects (P1-1B2008-43, P1-1B2010-49).

Contents

List of Figures

List of Figures

List of Tables

List of Algorithms

Chapter 1

Introduction

This chapter first describes the research context of this thesis. Then, the motivation, the objectives, and the main contributions of this work are presented. The chapter concludes with the organization of the rest of the thesis.

1.1 Research Context

In the last decade, the amount of data published on the Web has increased at an amazing rate. Data are published on the Web by individuals and institutions in order to be consumed by other potential users. In the beginning, most searchable data were published on web sites and were reachable by web search engines (e.g., Google, Yahoo, etc.). However, in last years, the ever growing amount of data has made unfeasible to publish all data on simple web pages. Nowadays, data are stored in "*containers*", accessible through the Web, and usually self-described in order to provide information about the data they contain. The information about the "*container*" and about the data it holds is known as *metadata*. Data containers available on the Web are called in this work *web resources*. A web resource is any application, information source, service or site that can be identified, named, addressed or handled in the Web or in any networked information system. A web resource usually provides functional and processable metadata describing its functionality, the data it holds and other features relevant to its discovery and processing. Examples of web resources are: web services providing relevant data to users, web services executing a specific programmatic function on data, databases or datasets accessible through RESTful services, and a web site containing relevant data and metadata describing them among others.

In the last years, web resources have become very popular in many domains. However, the huge number of resources and their decentralized distribution over the Web hamper their discovery by potential users. In order to make the web resource discovery easier to users, many web resource registries have been created in last years. A web resource registry is a repository of web resources metadata which provides a search engine to retrieve relevant resources to users' requirements. A web resource registry does not host the web resources, but their associated metadata and a link to the web resource provider's site. Some early, web resource registries performed the discovery on a set of predefined fields, e.g., input data type or resource type, whose terms are usually predefined in a taxonomy, e.g., BioMoby [108] and BioRegistry [30]. The use of specific fields limits the specification of user's requirements and, in consequence, the discovery of the most relevant resources becomes a tedious and ambiguous process. Currently, web resource registries consider all the web resource metadata available on the registry, e.g., textual descriptions, tags and categories. Recently, the web has become into what is known as the *social web* in which the user is the main source of information. This tendency has also been shown in web resource registries, and many current web resource registries allow any user to provide useful information about a registered resource with the aim of providing as much information as possible to better characterize it. These registries are commonly known as *open registries*.

In this thesis, web resource discovery is treated as an Information Retrieval [8] problem. Nowadays, IR is being addressed by the use of semantics [5; 53]. Semantics can be defined as the study of meaning, i.e., it studies the relation between signifiers such as words, phrases, signs and symbols, and that they stand for. Introducing semantics to the web leads to the Semantic Web, defined by Tim Berners-Lee as "a web of data that can be processed directly and indirectly by machines" [14]. The Semantic Web enables machines to interpret, combine and use data on the Web. The core of the Semantic Web is the computer-understandable description of resources, i.e., resources are annotated with computer-processable metadata. Semantic annotations enrich unstructured and semi-structured data with additional data formally described in external knowledge resources (ontologies) and understandable by computers. In this work, semantic annotation of resources metadata improves their discovery since it formalizes the information about the resource, and it reduces heterogeneity and ambiguity. So, considering the web resource discovery as a semantic-aware problem, it can take profit from both the techniques and tools developed on the classic IR research

field and those developed for the Semantic Web.

1.2 Motivation and Objectives

This thesis is focused on the Life Sciences domain, which involves the scientific study of living organisms such as plants, animals and human beings, as well as related considerations like bioethics. In this domain, researchers and institutions publish their data on the Web as well as resources and tools to manage them. Research in Life Sciences depends on the integration of large, distributed and heterogeneous web resources. The discovery of the most appropriate web resources to solve a given research task is still a complex research question.

In Life Sciences, open web resource registries have been gaining popularity due to the valuable social information provided by the Life Sciences community. Examples of open registries in Life Sciences are BioCatalogue [15] and myExperiment [38]. Unfortunately, discovery on these open registries suffers from the same drawbacks as the general-purpose registries, made even worse by the high level of heterogeneity of data in this domain.

Current open registries in Life Sciences do not provide much assistance to users during the discovery process. For the specification of users' requirements, most registries provide two types of search: keyword-based search and browsing through categories or tags. In the former, the user has to determine a set of keywords that best describe her requirements and that appear in resources metadata in order to make possible the matching. In the latter, the user has to select categories or tags for specific filters, limiting thus the specification of the user's requirements. In this type of search, the quality of the discovery results depends on the user's domain knowledge and on the coverage and specificity of the taxonomy of categories or tags. Then, at the end of the discovery, in most open registries in Life Sciences, the user is provided with a set of resources without any information about their relevance to her requirements.

Another drawback of current open registries is the high dependency of the discovery process on the characteristics of the resources metadata, i.e., vocabularies, format or structure, since most open registries rely on string matching techniques to perform the resource retrieval.

So, in conclusion, the discovery of the most appropriate web resources in Life Sciences given a user's requirement is hampered in current open registries by: *(i)* poor

representation of user's requirements, *(ii)* low assistance to the user, and *(iii)* discovery heavily dependent on the characteristics of the resources metadata.

The main goal of this thesis is to improve the discovery of the most appropriate web resources for a specific user's requirement by addressing the limitations of current web resource open registries in Life Sciences. In the proposed discovery process, the user is assisted during all the process, from the requirements specification until the selection of the most appropriate web resources, in order to finally select the most suitable ones. To achieve this goal, some aspects have to be considered.

First, the requirements specification is crucial in the discovery of the most suitable web resources. Therefore, the requirements specification must represent as best as possible the user's information needs and, to achieve that, the user must not be limited to specific vocabularies or to specific search fields. In contrast, the user has to be able to provide a rich description of what she really needs.

However, a richer requirements specification is not enough if the discovery is restricted to specific fields or to specific data. The discovery process must be driven by the user's requirements, and it must not depend on the characteristics of the available metadata. It has to consider as much information as possible from the metadata and, to achieve that, all the available metadata must be automatically processed.

Finally, in order to assist the user in the selection of the most appropriate web resources, the retrieved resources have to be ranked according to their relevance to the user's requirement, that is, how well each web resource fulfills the user's information need.

Taking these requirements under consideration, in this thesis we state the following hypothesis:

Hypothesis 1.2.1. *The normalization of data using knowledge resources allows the reconciliation between user's information needs and the available web resources. Consequently, it improves the discovery and ranking of resources, providing the user with the ones that best fit her requirements independently of the characteristics of data.*

1.3 Contributions of this Thesis

The main contributions of this dissertation are:

- Web resource discovery has been a research field for years. This thesis surveys the main architectures and techniques proposed for the discovery of web resources.

This thesis is focused on the Life Sciences domain, so that it also presents the main characteristics of the web resource discovery in Life Sciences as well as a brief description of the most popular web resource registries in this domain.

- The huge number of web resources, their distribution, and their heterogeneity hinder their discovery to potential users. Web resource registries allow users to discover registered resources but, unfortunately, they do not provide much assistance to them. This thesis proposes a discovery process that assists the user during all the process, from the requirements specification, in which the user is not limited by the use of specific vocabularies or taxonomies, until the selection of the most suitable resources, where the user is provided with relevant information about the resources that helps her in the selection of the most appropriate resource for her information needs.

- Discovery in current open registries depends heavily on the characteristics of the metadata, which is highly heterogeneous. In order to alleviate the heterogeneity and ambiguity in vocabularies and formats, we propose a normalization process that semantically annotates the data involved in the discovery process with domain knowledge resources. In this way, the data, specifically textual descriptions, become computer-processable and the heterogeneity and the ambiguity issues are alleviated.

- Textual descriptions contain implicit information relevant to the resource characterization that traditional search engines do not identify. We propose a knowledge extraction process that automatically identifies relevant information about the resource features implicitly described in textual descriptions.

- As mentioned before, the discovery must be independent of the characteristics of the data, and it must retrieve the resources that are supposed to be relevant for the user's requirements independently of how they are described. The IR model proposed in this thesis bases the discovery of the most appropriate web resources on their implicit semantics in order to reduce that dependency.

- Finally, in order to assist the user in the selection of the most suitable resources, the IR model provides a similarity function that allows the system to provide the user with a ranked list of the retrieved web resources, according to their relevance to user's requirements.

- We present BioUSeR, a prototype that illustrates the usefulness of the approach proposed in this thesis.

1.4 Organization

This thesis is organized in eight chapters (including this one). A brief summary of each chapter is shown below.

1.4.1 Second Chapter: Web Resource Discovery

This chapter introduces web resource discovery as a challenging task for users looking for web resources to fulfill their information needs. The chapter is divided into two differentiated parts: first, it describes the main features of a discovery system, explaining the different architectures and the different discovery techniques and, then, it focuses on the web resource discovery in the Life Sciences domain. This latter part describes the characteristics of the domain that have to be considered during the discovery, and it presents a brief description of the characteristics of current discovery tools. Finally, it surveys the most popular systems for the discovery of web resources in Life Sciences.

1.4.2 Third Chapter: Semantic Discovery of Web Resources in the Life Sciences

This chapter presents a framework for the discovery of the most appropriate web resources given a user's requirement in the Life Sciences domain. It describes the main characteristics of the proposed approach and how the main limitations of current approaches have been addressed.

1.4.3 Fourth Chapter: Normalization

This chapter addresses the dependency on the characteristics of the metadata and proposes a normalization process based on the semantic annotation of data and knowledge extraction techniques. First, it describes the state of the art of semantic annotation and, then, the semantic annotation process as well as the resources used in the annotation process are explained. Then, it describes knowledge extraction techniques in order to automatically identify relevant information about the resources that improve their

characterization and, consequently, improve their discovery. Finally, it describes how this normalization process is applied to the data involved in the discovery process.

1.4.4 Fifth Chapter: An IR Model for Web Resource Discovery

This chapter presents the IR model proposed in this thesis for the discovery of the most suitable web resources given a user's requirement. Firstly, the most common IR models in the literature are briefly described. Afterwards, the proposed IR model is presented describing its main elements. First, it explains the representation of the data considered in the discovery process. Then, it describes the relevance function proposed to estimate the suitability of a resource for a specific requirement. Finally, it describes the discovery method based on the normalization of data and that considers the relevance of the resources to rank them.

1.4.5 Sixth Chapter: Experiments

This chapter shows the results of the experiments carried out to validate and justify the discovery process presented in this thesis. First, it presents the results of the experiments performed to validate some specific parts of the discovery process. Then, it presents global experiments to validate the quality of the whole discovery process. Moreover, for a further evaluation, we compare our approach with other retrieval models and with other popular discovery tools.

1.4.6 Seventh Chapter: The Prototype

This chapter presents BioUSeR, a prototype for the semantic discovery of web resources driven by user's requirements in Life Sciences. It shows two example use cases, which visualize the whole process done by a researcher using BioUSeR to discover the most suitable resources for her requirements.

1.4.7 Eighth Chapter: Conclusions

The last chapter recapitulates the main contributions and results, discusses the limitations of our proposed techniques, and suggests possible extensions and open research lines. Moreover, a list of papers published as the result of this thesis work is included.

Chapter 2

Web Resource Discovery

In recent years, the number of web resources available over the Web has increased considerably. Due to the large amount and the heterogeneity of web resources, finding a suitable resource with respect to user's requirements is a challenging task, which is called *web resource discovery*.

Web resource discovery is a general-purpose problem that varies depending on the intrinsic characteristics of the domain in which discovery is performed. This thesis is focused on the Life Sciences domain, in which resource requesters are researchers that have information needs during their research which may be fulfilled with existing web resources. Therefore, discovering the most suitable web resources for their requirements is crucial in their research activity. Discovery in Life Sciences presents the same challenges than in other domains, but the discovery tool has to take into account its particular domain characteristics, such as the data heterogeneity or the complexity of some Life Sciences tasks.

This chapter presents the state of art of web resource discovery systems. Specifically, Section 2.1 defines the process of web resource discovery and surveys the different architectures and techniques proposed in the literature. Section 2.2 addresses the discovery in a specific domain, Life Sciences. Finally, Section 2.3 provides some conclusions about the research done in web resource discovery.

2.1 Web Resource Discovery

Web resource discovery is the act of locating a machine-processable description of a web resource that may have been previously unknown and that meets certain functional

criteria. The objective of the discovery is to provide the user with the most suitable resources for her information needs.

The web resource discovery can be considered as a particular case of an Information Retrieval (IR) system [8], more specifically a Recommendation System [75], which is based on metadata rather than on document content analysis. Therefore, as any IR system, the web resource discovery depends on: *(i)* the user's requirements and how they are represented, for example through keywords, requirements specification models (like $i*$, UML or MAP) or specific query languages (SPARQL, SQL), *(ii)* the description of web resources and its representation, which can be as simple as a few words, a URI, or more complex metadata descriptors, such as a tModel (in UDDI), RDF, DAML-S and OWL-S statements, and *(iii)* the mapping function on which the discovery is based.

In the literature, most of the research done on web resource discovery has been mainly focused on the discovery of web services, since in last years there has been an explosion of this kind of technology. For this reason, in this section many of the presented systems are focused on web services.

Next, Section 2.1.1 reviews the different architectures for discovery tools, and Section 2.1.2 describes the mapping techniques used to perform the discovery of resources.

2.1.1 Discovery Architectures

In the literature, different architectures have been proposed for the discovery of web resources, which can be broadly classified as: centralized and distributed architectures.

2.1.1.1 Centralized Architectures

In a centralized architecture, the descriptions of web resources are stored in a central registry. A web resource registry is a repository of metadata describing the registered web resources (e.g., the data on datasets or the functionality of network-addressable services such as web services). A registry offers a standards-based mechanism to classify, catalog and manage web resources, so that they can be discovered and consumed by other applications. Moreover, the registry must describe the information provided by the resource, e.g., what a dataset contains or what a web service does. As depicted in Figure 2.1, providers publish their resources in a registry in order to make them visible to requesters. Then, requesters submit a query to the registry in order to retrieve suitable resources for their information needs. In these systems, a requester can be a

person or even a software application looking for resources to fulfill automatically an information need.

Figure 2.1: Discovery in a centralized architecture

UDDI (Universal Description, Discovery and Integration) [1] was proposed in 2000 as a standard method for publishing and discovering network-based software components (web resources and specifically web services) of a service-oriented architecture (SOA). UDDI provides a registry of web services and programmatic interfaces for publishing, discovering and managing information about the web services described therein. Web services in a UDDI based architecture are accessed for binding through UDDI Application Programming Interfaces (API). However, UDDI has not had the expected relevance and it has not become the universal registry of web services as it was expected to be. In fact, most UDDI providers have closed their UDDI registries in the last years, e.g., IBM, Microsoft and SAP in 2006 and Microsoft in 2010. Therefore, other solutions have been proposed.

Current registries contain the descriptions and the URL of web resources, and the web resources are kept in the providers' site. These registries behave as web resources catalogues, since providers publish the description of their resources and the catalogue make them visible to other users through its discovery capability. For example, WebServiceList[2], RemoteMethods[3], and WSIndex[4] are registries that allow the discovery of web services about general domains.

It is worth noting that, in the last years, registries of datasets are emerging in order to make visible data or data sources that are not reachable by the typical discovery methods. For example, Datahub[5] performs the discovery of a huge variety of datasets, ReStore[6] is a registry of educational resources, and the World Bank[7] gives access to data

[1] http://www.uddi.org
[2] http://www.webservicelist.com
[3] http://www.remotemethods.com
[4] http://www.wsindex.org
[5] http://thedatahub.org/
[6] http://www.restore.ac.uk/
[7] http://data.worldbank.org/

services about social topics such as education, finances, gender among others. Another type of registry is Wikipedia[1], which provides access to data that are described with well-defined metadata and which has its machine-processable version in DBpedia[2]. All these systems provide users with sources that are assumed to contain the data that fulfill their information needs.

However, centralized architectures present some drawbacks. First, registration of resources is voluntary and, therefore, if the providers do not publish their web resources, requesters will not be able to find them. Then, the central registry usually becomes a bottleneck of processing, presenting problems of performance and scalability. There have been proposals to build distributed (replicated) UDDI to overcome this bottleneck [33; 102].

2.1.1.2 Distributed Architectures

In distributed or decentralized architectures, web resources descriptions are usually stored at the provider's site and they are gathered by a web crawler or search engine.

A web crawler is a program that browses the World Wide Web in a methodical, automatic manner. Search engines use web crawling as a means of providing up-to-date data and, in this case, as a means of retrieving web resources descriptions.

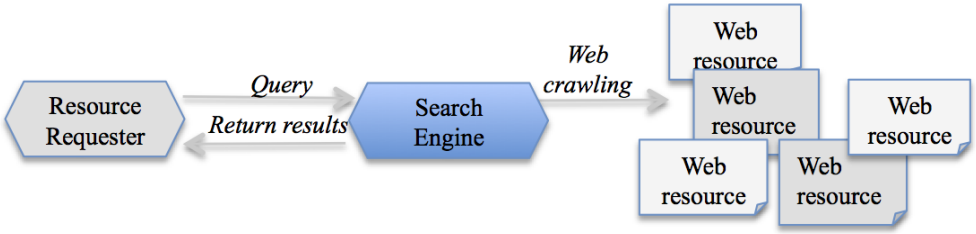

Figure 2.2: Discovery in a decentralized architecture

However, most crawlers of current search engines have been developed for web pages and not for web resource descriptions. For example, discovering web pages and web services presents significant differences that affect the final results. First, web services are usually described with a WSDL file that does not contain much textual information about what a service offers. Instead it contains complex technical information represented by non-standard structures in XML format. In contrast, web pages are built

[1] http://en.wikipedia.org/wiki/Main_Page
[2] http://dbpedia.org/About

with standard HTML files that contain a lot of textual information and links to external documents that enrich the services information. Therefore, current general-purpose search engines crawl web services descriptions assuming that they contain textual information that can be indexed or treated in the same manner as web pages, but most web services descriptions do not contain an adequate level of information. Moreover, some search engines, such as Google, rank results using the link structure, and special properties of HTML documents not applicable to WSDL files.

With respect to datasets, general-purpose search engines cannot distinguish database interface pages from documents mentioning them, and consequently they reveal themselves rather inefficient for discovering databases. Thus, these search engines return a mixture of scientific articles, web sites, tools, departments and people in a manner that makes extracting useful information very difficult. Moreover, many datasets are in the deep Web (web content not indexed by search engines and not linked to other pages), so if they are not explicitly published in any registry, they cannot be discovered by researchers.

Therefore, neither the identification of suitable resources through pure keyword extraction nor the relevance ranking based on HTML characteristics, such as hyperlinks and title tags, provides much of a use in a web resource scenario.

To overcome these limitations some new formats have been proposed to enrich the web pages information, e.g., microdata[1] and RDFa[2]. These specifications add semantics to the content of web pages by using machine-readable tags.

With respect to web resources, there are some approaches that enrich WSDL descriptions with information gathered from external resources [1] in order to provide more useful information about the service in the WSDL file. Others try to complement each architecture's strengths by combining search in centralized registries with search engines [3].

2.1.2 Discovery Techniques

Independently of the architecture that determines the location of web resources metadata, the discovery techniques can also be classified according to the adopted mapping function as either functional or non-functional based methods.

The functional methods base the discovery on the functionality provided by the web

[1] http://www.w3.org/TR/microdata/
[2] http://www.w3.org/TR/rdfa-core/

resource, which can consist of simple tasks such as data retrieval, a transformation of data, more complex tasks such as prediction methods, or domain specific algorithms (e.g., complex mathematical calculations). The functionality, which is usually described in web resources metadata, plays a decisive role in the discovery of web resources driven by users requirements. On the other hand, non-functional methods consider other features of the resources that are related to how the resource is supposed to be, e.g., quality and performance.

Next, two types of functional methods and the most popular non-functional methods are described.

2.1.2.1 Syntactic-based Functional Methods

Syntactic-based methods rely on the matching, either strict or partial matching, of the words in the user's requirements specification (i.e., query) and the words in the web resources metadata.

In the registries that use syntactic methods, the requirements specification usually consists of either a set of keywords or a set of categories taken from a concept hierarchy, which describes specific metadata of resources such as the functionality or data types of the resources.

In keywords-based discovery, many registries adapt the Vector Space Model (VSM) [97] for representing both user's requirements and web resources metadata. In VSM, descriptions having similar content are represented as vectors located near in the space. Then, the discovery of web resources is a one-to-many matching technique to find the nearest neighbors in a vector space. Therefore, the similarity between requirements and web resources can be estimated with measures that calculate the similarity between vectors. The most popular similarity measure for SVM is the cosine coefficient [96]. The main drawback of keyword-based search is that it might not discover relevant resources described with a different vocabulary from the query (e.g., synonyms).

With respect to category-based search, the user selects the category that best describes her requirements, usually after navigating through a hierarchy of well-defined categories; then, all resources annotated with that category are retrieved. The quality of the results depends on the correct assignment of categories to the resources and also on how well the category describes the user's requirements. The assignment of an irrelevant category to a resource might hide it in a category-based discovery.

Apart from keyword-based and category-based discovery, more sophisticated ap-

proaches have been proposed in the literature which apply syntactic matching on specific data. For example, there are approaches that use the Query-by-example (QBE) method, in which the query consists of either an example of the data processed by the service [15] or of a description of the service skeleton, which involves keywords, operation and parameter names [26]. Other approaches consider the structure of the service in order to do the matching, for example, approaches that calculate the similarity taking into account the service interface [106]. However, this method requires that the providers assign meaningful names to input/output parameters of web services operations. Moreover, it has to deal with ambiguity problems such as operations of a specific functionality with different signatures, which makes signature matching difficult, or multiple data types assigned for the same parameter.

2.1.2.2 Semantic-based Functional Methods

In the last years, semantics have been gaining importance in the discovery of web resources, since they facilitate the automation of resource related tasks such as discovery, interoperability, execution or composition.

Semantic matching is a technique used to identify information which is semantically related, that is, to search correspondences by mapping semantic descriptions (concepts), stored in knowledge resources, not by mapping words as in syntactic matching.

Semantic discovery relies on: *(i)* the semantic annotation model and *(ii)* the applied discovery method.

Semantic annotation models. The World Wide Web Consortium (W3C)[1] recommends three languages of different expressivity for the semantic representation of data. RDF[2] is the least expressive and it represents data in the form of triples $<subject, predicate, object>$. Then, RDF Schema[3] (RDF-S) extends RDF with mechanisms for describing groups of related resources (classes) and the relationships between them (properties). Finally, OWL[4] (Web Ontology Language) facilitates greater machine interpretability of web content than that supported by XML, RDF and RDF-S by providing additional vocabulary along with formal semantics based on description logics [7].

[1] http://www.w3.org
[2] http://www.w3.org/RDF/
[3] http://www.w3.org/TR/rdf-schema/
[4] http://www.w3.org/TR/owl-features/

Microdata[1] and RDFa[2] have been proposed for the annotation of the content in web pages. Microdata is a simple specification that allows to describe the content in the HTML document by annotating HTML elements with a supporting vocabulary using name-value pairs. RDFa enriches HTML, XHTML and XML documents with rich metadata by using RDF triples.

There are also formats to annotate semantically web resources descriptions. Some of them have their own semantic models and formal languages to describe web services semantically, e.g., OWL-S[3] and WSMO[4]. OWL-S is based on OWL and it allows to describe what a service does and how a client can use and interact with it. WSMO provides a conceptual framework and a formal language for semantically describing all relevant aspects related to semantic web services. Other approaches add semantics directly on the description files of web resources. For example, SAWSDL[5] provides mechanisms by which concepts from the semantic models can be referenced from within WSDL components as annotations.

With respect to the annotation process, in the last years several attempts have been made to semantically annotate texts. Depending on the human intervention, the tools can be classified as: manual, semi-automatic or automatic. Manual tools assist the user in the annotation process, and rely on the knowledge and will of the users to annotate entities in text. Examples of manual tools are Annotea [50], NOMOS [70] or CREAM [43]. Other tools automate some of the stages of the SA process using, for example, user-defined rules such as Melita [24] and [10], or using a bootstrapped information extraction process based on the redundancy of the Web as KnowItAll [32]. Although results of supervised SA approaches are usually satisfactory, the intervention of a curator in the annotation process supposes a huge cost of time and resources. Therefore, automatic and unsupervised systems are the most desirable ones, though their performance is usually worse than manual systems. Examples of automatic and unsupervised systems are: SemTag [31], which performs automated semantic tagging from large corpora based on the Seeker platform for text analysis, and Pankow [23], which uses syntactic patterns to mark-up candidate phrases in web pages without having to manually produce an initial set of marked-up web pages.

[1] http://www.w3.org/TR/microdata/
[2] http://www.w3.org/TR/rdfa-core/
[3] http://www.w3.org/Submission/OWL-S/
[4] http://www.w3.org/Submission/WSMO/
[5] http://www.w3.org/2002/ws/sawsdl/

Discovery methods. Semantic discovery methods depend on the available web resources semantic metadata. The most simple methods are those that estimate semantic similarity between functional description of resources and user's requirements, using the information in lexical resources such as WordNet, and considering simple relationships between terms such as the synonymy. More sophisticated methods consider the underlying relationships between concepts in an ontology. Techniques such as ontology linking or Latent Semantic Indexing [28] calculate the similarity between the semantics of the resource metadata and the semantics of the users' requirements, taking into account the relationships between ontological concepts.

Depending on the information taken into account, semantic discovery methods can be classified as: *(i)* functional semantics methods and *(ii)* context-based methods.

In functional semantics methods, the degree of matching between web resources and users' requirements is computed based on the matching between concepts and functional constraints describing the functionality. Even though functional semantics provides a unified way to semantically describe a functionality by the resource provider and requesters, there is a lack of capability in expressing functionality. WSMO is one of the few frameworks that promotes a goal-based approach for semantic web resources, and [111] proposes a unified way to describe the web resource functionality and resource requests using functional semantics.

On the other hand, context-based methods take into account additional information about the context of the description (both in web resource metadata and users' requirements) apart from the functionality of the resource. The information about the context helps to enrich the specification of users' needs and the characteristics provided by a resource.

2.1.2.3 Non-functional Methods

Non-functional properties of a resource describe other aspects apart from its functionality, e.g., execution or usage qualities. These properties are rarely used in the discovery of web resources, since discovery is usually driven by the functionality that must be fulfilled, but they are often used in the ranking of the results, since they help to distinguish resources with similar functionality but with different qualities that may be relevant for users' requirements.

[90] provides a list of QoS (Quality of Service) parameters that can be classified as: *(i)* runtime related QoS such as scalability, availability, performance and so on,

(ii) transactional-related QoS such as integrity, *(iii)* configuration management and cost related QoS, such as cost, stability or completeness, and *(iv)* security related QoS such as authentication or data encryption. In recent years, non-functional properties about popularity and usage of the resources are gaining importance due to the success of social information on the web. The reputation of a web resource may determine the final selection since opinions of other users are highly valuable. [29; 110] consider QoS during the discovery, and others such as [4] take QoS also into account in the ranking of the discovered results.

2.2 Web Resource Discovery in Life Sciences

In recent years, the research activity of Life Sciences community has produced a huge amount of data as well as many resources and tools, most of them now available on the Web, to process those data.

A web resource in Life Sciences is anything that can be unequivocally identified and that provides a specific functionality required by researchers in which biological data are involved. A web resource can be a web page that visualizes relevant biological information and metadata describing it, databases containing biological data or web services that perform algorithms that process and transform biological data, e.g., a web service that aligns two DNA sequences.

In Life Sciences, researchers use discovery tools in order to find the web resources that are the most appropriate for their information needs. A researcher initiates the discovery by describing the information she needs, either data stored in a dataset or the result of processing some data. The discovery must provide the researcher with a resource or a set of interconnected resources in case required data are provided by the execution of several resources (workflows).

However, the discovery of the web resources is a challenge for Life Sciences researchers due to the huge amount of available web resources and to their heterogeneity and decentralized distribution. Moreover, there is not any widely-accepted standard to represent data and, therefore, each resource provider represents the data with different vocabularies which makes the discovery and the integration of web resources a complex task. In last years, the issues of web resource discovery in Life Sciences have been addressed by many systems with the common goal of assisting researchers in the selection of the most appropriate resources.

Most of the discovery systems are based on centralized architectures. Most resources are available on registries or catalogues in which providers publish their resources with useful metadata about the functionality the resource provides, the data involved, and additional features such as parameters or constraints. In Life Sciences, there are systems focused only on the discovery of datasets (e.g., DAS [88], MIRIAM [49], and BioRegistry [30]), others only deal with web services (e.g., BioMoby [108], SADI [109], BioCatalogue [15], and myExperiment [38]), and a few consider any type of web resources (e.g., SSWAP [36], Bioinformatics Link Directory [18], and ExPASy [74]).

Regarding the use of general-purpose search engines, apart from the limitations pointed out in Section 2.1.1, search engines are quite sensitive to the domain specific data and to their heterogeneity. Some domain specific crawlers have been developed for discovering biological databases. For example, BioSpider [54] and ACHE [9] implement strategies for filtering web forms in order to retrieve database interface forms. However, the resulting collections of links are poorly indexed, and do not allow efficient discovery.

Apart from the type of resource, registries can also be distinguished by: *(i)* their discovery method and *(ii)* the metadata considered during the discovery.

Considering the discovery method, many registries base the discovery on syntactic methods (see Section 2.1). In these registries, results heavily depend on the specification of the user's requirements due to the high heterogeneity of data [62]. As [37] claimed, standardization of data in Life Sciences is unlikely to happen soon and, therefore, other normalization techniques such as the use of semantics have to be considered.

In the last years, semantics have been gaining relevance in the web resource discovery in Life Sciences with the aim of addressing the previous issue. Nowadays, there is a large number of KRs for annotation (e.g., BioPortal [72]). However, providing semantic annotations manually is a tedious and time-consuming process that requires providers to know the adopted knowledge resources. Currently, there are automatic tools that semantically annotate biomedical texts, e.g., BioPortal, EAGL [93], MetaMap [6] and Whatizit [91]; however, they are not suitable for the annotation of web resources metadata since they do not cover all the terminologies used in the Bioinformatics domain.

Recently, there have been several efforts to publish the semantic representation of biomedical information available on databases or on other resources such as scientific articles. For example, Bio2RDF [11] and Linked Life Data[1] represent the content of biomedical sources in RDF with the aim of helping the process of bioinformatics

[1] http://linkedlifedata.com/

knowledge integration. With respect to scientific articles, AO [22] provides a common model for document metadata derived from text mining and manual annotation of biomedical scientific papers and that can be published as Open Linked Data on the Web. Moreover, in the last years, there have been many efforts with the aim of making the use of semantics easier for users. For example, BioPortal and [67] provide semantic functionalities that can be used, among other applications, for the annotation of web resources metadata. Others aim to help researchers to consume and integrate the data from those databases available in RDF. For example, BioQueries [34] is a wiki-based portal to encourage users to share their experiences with Biological Linked Data by publishing their SPARQL queries in the portal.

Regarding the semantics in current web resource registries in Life Sciences, few registries provide semantic representation of resources metadata. Some registries support a very limited use of semantics by allowing providers and requesters to use terms from an ontology as tags of a resource. For example, BioCatalogue has the myGrid ontology as the reference vocabulary to provide tags but, unfortunately, users hardly ever use terms conforming to that ontology. Others use BioMoby as the reference vocabulary, but they present similar drawbacks. Lately, some registries have incorporated semantics allowing to store the resources metadata in RDF and to specify semantic queries. There are registries that combine syntactic and semantic (SPARQL queries and query graphs) search, like myExperiment [38] and SSWAP [36], whereas others are based mainly on semantics, e.g., SADI [109]. In these semantic web resources registries, the discovery and composition are easier than in the other registries due to the benefits of semantics. However, they require the users to provide the appropriate semantics when publishing the resource, and when looking for a suitable one, which supposes an extra effort for them since they have to know the knowledge resources and semantic-aware technologies such as SPARQL. Moreover, the semantics are usually still related to specific metadata, and rich free-text descriptions remain unannotated.

With respect to the metadata considered during the discovery, web resources metadata usually consist of well-defined fields, like categories and tags usually describing specific features of the resources, and rich textual descriptions. Searching through browsing only considers well-defined metadata, whereas keyword-based search usually considers all the available metadata in the string matching process.

Recently, due to the relevance that social information has gained in almost all domains, some registries allow their users to provide metadata about the resources, for

example: descriptions, tags, categories, additional information such as publications, comments, data examples or even information about the resource popularity with ratings or counters of the times the resources have been viewed or downloaded. These registries are commonly known as *open registries*.

Here, next in this section, the most popular discovery tools in Life Sciences are briefly described considering the features described in this chapter. Firstly, registries of data services are presented, then, registries of web services and, finally, registries that allow the discovery of different types of resources.

2.2.1 Data Services Registries in Life Sciences

Data services registries aim to identify and annotate biological data in order to make data integration easier. Table 2.1 shows the main characteristics of each one of the surveyed data services registries.

- **DAS** [88]. Distributed Annotation System is a widely-accepted communication protocol used in the exchange and integration of biological data. DAS is used to annotate many different kinds of biological entities, e.g., genome sequence, protein sequence, molecular structure and so on. DAS was designed as a lightweight system for integrating data from a number of heterogeneous distributed databases, and it allows the discovery of available DAS sources via a web page, or as a machine-readable XML that can be used directly by DAS client programs. The sources in DAS registry are described by a title, a short free-text description and machine-readable metadata describing the capabilities, the type of the resource and the data types. The discovery is based on a textual description and on the filters defined on the specific metadata such as the capabilities, the type or the organism. As of March 2013, the DAS registry contains 1579 sources provided by 53 groups in 17 countries.

- **BioRegistry** [30] is a registry of biological databases, in which metadata are attached to biological databases organized in a flexible and structured manner, enabling knowledge modeling about biological databases and advanced discovery capabilities. The registry is automatically generated from a publicly available list of biological databases, the Molecular Biology Database Collection[1] published in Nucleic Acid Research (NAR). It associates metadata, automatically extracted

[1] http://www.oxfordjournals.org/nar/database/c/

from NAR, to the databases, e.g., description, input and output options, MeSH terms and categories. The repository allows browsing with MeSH terms and categories, and searching by specifying the name or id of the required resource, in case the user knows it, or with a set of keywords matching the databases description using the boolean operators AND and OR. As of March 2013, it has 1221 registered databases.

- **MIRIAM** [49]. Minimum Information Required in the Annotation of Models (MIRIAM) registry is a catalogue of biological data collections, for each of which extensive metadata are recorded. Its aim is to assign Uniform Resource Identifiers (URIs) to uniquely identify any record in a collection. Discovery can be made by exploring the list of databases, a list of tags, or by strict string matching search. As of March 2013, it has 414 data collections and 506 resources providing access to these collections.

Registry	**Resources**	**Metadata**	**Discovery**		**Semantics**	**Number of**
			Browsing	**Searching**		**Resources**
DAS	DAS services	Description, status, capabilities, types, data types	Filters on metadata on free-text description	String matching	Controlled vocabulary	1579
BioRegistry	NAR databases	Name, description, citations, categories, MeSH terms	MeSH terms, categories	Keywords, name, id		1221
MIRIAM	Data collections	Description, namespace, website, categories, usage examples	Names, tags	Sentence		414 data collections, 506 resources

Table 2.1: Most popular data service discovery tools in Life Sciences

These registries contain well-defined metadata of services that provide access to biological datasets and databases. The discovery can be made by: *(i)* keyword-based search and *(ii)* browsing through categories (BioRegistry), tags or names (MIRIAM) or filters on specific fields (DAS). None of them uses semantics to formalize the services metadata, nor provides the user with a ranked list of results.

2.2.2 Web Services Registries in Life Sciences

Web services registries provide a common interface to discover Life Sciences web services independently of their functionality. Table 2.2 shows the main characteristics of the surveyed web services registries.

- **EMBRACE Service Registry** [85] is a Life Sciences web service registry with built-in service testing developed in the EMBRACE (European Model for Bioinformatics Research and Community Education) project, together with EDAM [86] (an ontology for describing Life Sciences web services) and BioXSD [51] (a schema for exchanging data between services). Each registered web service is described with tags, a WSDL file, a textual description, and its status. One important and useful characteristic of this system is that each entry includes live test data. Although efforts at semantics have been done in the EMBRACE project, the registry provides a syntactic discovery method based on the string matching of query keywords. This registry is considered as the prelude to BioCatalogue, and all its services have been also registered in BioCatalogue. Currently, it has 822 web services but the last updates were done in 2010.

- **BioMoby** [108] is an open-source research project whose aim is to produce a simple, extensible registry to enable discovery, representation, integration and retrieval of biological data from widely disparate data hosts and analysis services. The components of BioMoby are: MOBY Services (bioinformatics software tools), MOBY Objects (input and output data for the services) and MOBY Central (a register holding the input/object types of all registered resources, their URL and their service types). The discovery of BioMoby services in MOBY Central is based on their input, output, service type or authority by using the object and service ontologies. The use of these ontologies facilitates the matching of web services consumers, who have in-hand BioMoby data, with service providers, who claim to consume that data-type (or some compatible ontological data-type) or to perform a particular operation on it.

  Currently, there are several BioMOBY central repositories providing BioMoby services. Moby 2.0/CardioSHARE [104] is a RDF-based system whose goal is to provide a higher level of functionality and reasoning capabilities. In this approach, data is interchanged in RDF and queries are expressed in SPARQL. This project expects web services to be able to consume and produce RDF. Nowadays, it is

based on the use of Bio2RDF [11], that aims to enable the use of standard OWL reasoning techniques to improve service discovery. CardioSHARE can access to SADI services in response to SPARQL queries. It has more than 1500 web services (as of March 2013).

- **Magallanes** [68] is a library of algorithms aimed at discovering Bioinformatics web services. The search is based on a Google^{TM}-like approach, in which the user keywords are matched to metadata descriptions improved by the *Did you mean...?* algorithm, which helps the user to build the query. Search can be performed on data type, service type, service and operation fields, and it supports boolean operators. Moreover, Magallanes provides a way of composing compatible services into workflows.

  Magallanes provides the user with a ranked list of retrieved services based on the similarity between the keywords and the web service metadata.

  Currently, Magallanes is used as a discovery engine in other integration tools such jORCA [60] and MOWServ [89], in which it has access to more than 700 web services registered in the INB repository.

- **MOWServ** [89] is a bioinformatic platform developed by the Spanish National Institute of BioInformatics (INB) that provides integrated access to the services in the INB repository. It provides two types of web service discovery: *(i)* browsing through the BioMoby taxonomies for services and data types , and *(ii)* discovery based on the input data. For more advanced searches, it uses the Magallanes system. Currently, it has access to more than 700 web services registered in the INB repository (as of March 2013).

- **SADI** [109] is a framework that uses standard-compliant Semantic Web Service design patterns that simplify the publication of services and their subsequent discovery in domains such as Bioinformatics. Providers have to follow SADI conventions to publish their services, e.g., all services consume and produce RDF instances of OWL classes, and users have to use semantic discovery tools to query the semantic metadata, for example, through SPARQL queries. SADI can be used as framework in other web services registries, e.g., SADI is currently used in CardioSHARE as platform to solve SPARQL queries. Currently, in its own SADI

registry[1], there are 688 web services registered (as of March 2013).

- **BioCatalogue** [15] is a Life Sciences registry that provides a common interface for registering, browsing and annotating Life Sciences web services. Curation of information about web services is open to any user in the Life Sciences community and uses a combination of free text, tags, ontology terms and examples values to describe the service functionality, the type of biological data and data formats that the service accepts or returns among other additional features. These annotations are manually provided not only by the resources providers, but also by users. Moreover, some information is gathered by some monitoring and usage analysis data obtained automatically by BioCatalogue servers. However, most of these annotations are expressed as free text without following any controlled vocabulary. The service discovery is mainly based on the keyword search and browsing over different aspects. The keyword-based search consists of the string matching between user's keywords and services metadata. Browsing can be performed over: service type (SOAP/REST), provider, submitter, country and category, which is the most useful filter since it is based on a taxonomy about services functionality. To enhance its accessibility and usability, BioCatalogue is indexed by search engines such as Google^{TM}. It also provides a programmable API which is used by third-party applications such as Taverna. Currently, BioCatalogue contains 2332 registered web services from 165 providers from 31 different countries.

- **myExperiment** [38] is a Life Sciences repository, developed in the same project as BioCatalogue, whose main resources are workflows but other research objects can also be registered in it. The workflows are annotated with tags, a textual description, object types and information about the provider. Moreover, myExperiment provides additional information about non-functional features of the resources such as the number of times the resource has been viewed or downloaded, the number of users that have defined the resource as their favorite, and a rating scale that reports the opinion of users about the quality of the resource. This information is highly valuable for the user when selecting the most suitable web resource.

  The discovery is based on filters over well-defined fields, e.g., type, tags, user, license and so on, and on a keyword-based search that considers all the metadata

[1] http://sadiframework.org/registry/services/

available about the resource. Currently, myExperiment publishes all its public data on RDF, which can be queried with SPARQL. As of March 2013, myExperiment provides access to 2729 workflows, to 300 packs, and there are 8947 registered members.

- **Taverna** [73] is a workflow construction environment and execution engine designed to support *in silico* experiments developed by the European Bioinformatics Institute (EBI) and the University of Manchester. Taverna is part of the ^{my}Grid project, so it is aligned to BioCatalogue and myExperiment. It is able to build complex workflows, to execute them, and to display the different types of results. To build the workflows, the user has to search the appropriate web services, which can be available in Taverna or can be imported by their URL, which requires that the user knows the service or has performed a previous discovery with another discovery tool. Taverna performs web service discovery using string matching between the user's keywords and the web services metadata. Moreover, the user can also search services by the input/output data types on BioMoby services. Taverna, as of March 2013, provides access to more than 3500 resources.

Some registries restrict the registration to specific service architectures, e.g., SADI, whereas others accept different types of architectures, such as SOAP and REST, and the discovery is transparent to the service type. Current web services registries facilitate well-structured metadata about the registered web services that can be described by: *(i)* predefined values defined in taxonomies (e.g., the categories in BioCatalogue), *(ii)* values conforming to an ontology (e.g., the service type or data types in BioMoby and SADI), or *(iii)* free labels (e.g., the tags of EMBRACE, BioCatalogue and myExperiment). Moreover, most of these registries also contain textual descriptions of the services which are used for keyword searches.

With respect to semantics, there are registries that have their own ontologies to define the metadata, e.g., BioMoby and myExperiment, and others use third-party ontologies, e.g., SADI. Moreover, there are registries that represent the services metadata using RDF, which allows the use of SPARQL and semantic query graphs for resource discovery.

Registry	Resources	Metadata	Discovery		Semantics	Number of
			Browsing	Searching		Resources
EMBRACE	SOAP, REST, DAS, BioMoby	Tags, WSDL, description, status	Name, tags	Keyword matching		822
BioMoby (CardioSHARE)	BioMoby services	Service type, I/O types, description	Virtual graph	Service type, Object type, SPARQL	BioMoby ontology, RDF	1500+
Magallanes		Textual description, service type, data type		Keyword matching		700+
MOWServ	BioMoby, INB services	Service type, I/O types, description	Service type, data type	Keyword matching (Magallanes)	BioMoby, own ontology	700+
SADI	SADI services	I/O types with properties		SPARQL, Virtual graph	Third-party ontologies, RDF	688
BioCatalogue	SOAP, REST	Categories, tags, description, data examples, publication, citations, monitoring	Categories, tags, providers, submitters, countries	Keyword matching on all metadata		2409
myExperiment	Workflows	Tags, description, rating, version, viewed, downloaded, reviews, comments,	Type, tags, user, licence, group, wsdl, curation	Keyword matching, SPARQL	RDF, own ontology	2729
Taverna	Workflows, BioMoby, WSDL, BioMart, SoapLab	I/O BioMoby types	BioMoby	Keywords matching	BioMoby data types	3500+

Table 2.2: Most popular web service discovery tools in Life Sciences

2.2.3 Web Resources Registries in Life Sciences

Web resources registries allow the discovery of different types of web resources by providing a common discovery interface for all of them. Table 2.3 shows the main characteristics of the surveyed web resources registries.

- **SSWAP** [36]. Simple Semantic Web Architecture and Protocol (SSWAP) proposes an architecture, a protocol and a platform to semantically discover and integrate heterogeneous disparate resources on the Web. SSWAP proposes a unique canonical structure for all actors and activities; the same canonical struc-

ture that allows providers to describe their resources is the same structure for expressing queries, which is in turn the same canonical structure for phrasing service invocation, which is the same structure for representing results.

SSWAP architecture is based on five basic concepts: *Provider, Resource, Graph, Subject* and *Object*. Providers correspond to organizations that own and publish resources. Resources can be arbitrary resources like web pages, ontologies, and datasets; but they are primarily used to describe web services. The transformation performed by the service is described by the Graph concept, which defined a mapping from a SSWAP Subject (input) to a SSWAP Object (output). All this information is stored in RDF.

Resources are described by a Resource Description Graph (RDG) accessible by anyone via a simple HTTP GET. Users' requirements can be specified in three ways: using a keyword-based search through the web front-end (http://sswap.info) or programmatically engaging the SSWAP query service with a Resource Query Graph (RQG) or with a Resource Response Graph (RRG). The discovery using the RQG is based on the partial matching between a RQG and a RDG, considering specifications and generalization of concepts. The RRG is the graph returned by a previous service, facilitating in this way the interoperability between resources.

[36] stated that SSWAP had, as of 2009, more than 2400 web resources.

- **Bioinformatics Link Directory** [18] contains more than 2100 curated links to Bioinformatics resources, databases and tools, including all the databases and web services listed in NAR (Nucleic Acids Research) special issues, organized into 11 main categories. Each link is described by: a name, a textual description, categories, tags, publications (PubMed links), user feedback (rating), and the options to access or download it. It provides two ways of discovering resources: *(i)* browsing through categories and subcategories, and *(ii)* searching with keywords-based queries performing string matching within the title, the description and the tags of the link. As of March 2013, it has 163 resources, 620 databases and 1376 tools.

- **ExPASy** [74] is an extensible and integrative portal accessing many scientific resources, databases and software tools in different areas of Life Sciences, most of them provided by SIB (Swiss Institute of Bioinformatics). Resources are described by a name, a textual description (usually very short), scientific categories,

keywords from a controlled vocabulary, an URL and the status. ExPASy provides two types of discovery: *(i) Find resources* discovers databases and software tools using string matching on name, keywords, category and descriptions, or by browsing through the hierarchy of categories, and *(ii) Query databases* (cross-resource search) in which a text-based query is sent in parallel to a set of selected resources returning the number of hits and the link to the query results. As of March 2013, ExPASy registered 267 resources.

Registry	**Resources**	**Metadata**	**Discovery**		**Semantics**	**Number of**
			Browsing	**Searching**		**Resources**
SSWAP	Web services, data resources	Name, description, I/O types		RRG, RQG, keywords matching	RDF	2400 (year 2009)
Bioinformatics Link Directory	Resources, databases, tools	Categories, description, MeSH terms, tags, rating, PubMed links	Category matching filtered on: titles, description or tags	Keywords		163 resources, 620 databases, 1376 tools
ExPASy	Resources, databases, software tools	Categories, keywords, software types, status, description		Keyword matching, cross-resource database search		267

Table 2.3: Most popular web resource discovery tools in Life Sciences

To sum up, apart from the common discovery interface, some registries also provide specific discovery functionality for specific types of resource. For example, ExPASy describes all resources in the same manner, but it provides a specific search to query databases directly, apart from the general search.

In some registries, resources are described with external resources. Some of them formally describe the metadata through knowledge resources, like SSWAP, whereas others describe resources providing additional information about them, like Bioinformatics Link Directory links the resources to PubMed articles.

2.2.4 Discussion

The surveyed web resource registries present similar characteristics and, therefore, similar limitations that hinder the discovery of the most adequate web resources for specific user's requirements.

Firstly, current registries in Life Sciences limit the user in the specification of what she needs and, in consequence, the discovered resources may not be those expected by the user. The most popular way to specify user requirements is through keywords-based query, with which the search engines perform string matching over resources metadata. In the keyword-based search, the user has to summarize her information needs into a set of keywords that are supposed to be the best fittered for representing them. Moreover, these words must explicitly appear in the resources metadata, since most registries use string matching techniques, which hardly ever consider variants of words, neither synonyms nor related words such as hypernyms. Due to the heterogeneity of metadata descriptions that come from different providers, the recall of the discovery can be very low. Many registries also support browsing through categories, which are usually related to either the research task or to facet values. Browsing through categories also limits the user in the specification of her requirements, since she has to choose categories from a hierarchy, or values from a controlled vocabulary related to specific metadata. Both categories and filter-value pairs hardly ever describe accurately the user's information needs. So, in current registries the user is limited when describing her requirements since she has to make a double effort: *(i)* to know the vocabulary used to describe resources, and *(ii)* to summarize her information needs into a set of keywords or categories of that vocabulary. These limitations affect directly the discovery process since the discovered resources may not be as relevant or as precise as required and, therefore, may not achieve completely the user's requirements. Other registries (e.g., SSWAP, SADI) base the discovery on semantics and allow the users to provide a more formal specification of their requirements using semantic query languages such as SPARQL and graphs. However, although these languages provide a more precise description of the user information needs, the specification is more complex than in other querying techniques, and users have to be experts on the query language, which requires also an extra effort.

Another limitation of current registries is related to the characteristics of the available metadata and their role in the discovery process. The amount and the characteristics of web resources metadata depend on the facilities provided to the users by the registry to publish metadata and, also, to the users' will. In some registries, resources are described with well-defined metadata, i.e., structured metadata with values from a controlled vocabulary, which facilitate the automatic discovery when the discovery is based on the same specific metadata. However, web resources described only with well-

defined metadata are usually poorly described since these metadata are normally very specific and related to few specific aspects of the resource. For example, BioMoby allows describing the resources with their type and their input/output parameters. However, the search is based only on these specific fields, limiting the requirements specification. In most recent registries, the resources are described with well-defined metadata and with textual descriptions with richer information. These registries combine the search on the specific metadata with that based on the textual descriptions. However, textual descriptions are processed with string matching techniques, which do not identify relevant information implicitly described in the text such as the resource functionality and its features. Though current registries usually provide specific fields to specify those relevant features, evidence shows that most of this information is implicitly described in textual descriptions and, therefore, it is not taken into account in the discovery process, e.g., BioCatalogue and myExperiment. Therefore, the discovery in current registries depends heavily on the way resources are described, both structural (where the information is described) and lexical (the vocabulary used to describe the resources and the user's requirements). Semantic-based registries, like SSWAP and SADI, alleviates this dependency thanks to the formal description of metadata. However, the metadata are limited to specific fields and the semantics must be specified by the resource provider, which can be a hard and non-trivial task for providers.

Finally, as a result of the discovery process, the user is prompted with a set of resources that are supposed to fulfill her information needs, but they are hardly ever ranked on base to their relevance to the user's requirements. Few registries, e.g., Magallanes, rank the results according to the similarity between the resource description and the requirements specification, but they just count the number of matched words, without considering further relevance criteria. To the best of our knowledge, no registry considers the relevance of keywords during the ranking.

To conclude, the main limitations of current open registries in Life Sciences can be summarized as follows:

1. Low representation of user's requirements by current requirements specification techniques.

2. Dependency on the properties of the available web resources metadata.

3. Lack of identification of relevant information implicitly mentioned in textual descriptions which would be useful to characterize the discovery and, in consequence,

improve the results.

4. Low assistance to the user once provided the set of discovered results.

All these limitations turn the discovery of the most appropriate web resources into an imprecise and complex task. As a consequence, users end up searching for familiar resources since they know the names or how they are described, but which are not always the best suited for their requirements.

2.3 Conclusions

The discovery of the most suitable web resources given a user's requirement has become a challenge for users due to the increasing number of web resources over the Web and to their distribution. Moreover, the challenge of the discovery of web resources in Life Sciences is intensified due to the heterogeneity of data.

Current web resource registries in Life Sciences present some limitations that hinder the discovery of the most suitable resources given a specific user's requirement: *(i)* poor representation of user's information needs, *(ii)* high dependency on the characteristics of the available resources metadata, and *(iii)* low assistance to the user during the discovery process.

These limitations make that search engines of current registries do not always provide the user with the resources that best fit her requirements, maybe because the user has not described precisely what she really needs, or maybe because there is not any matching between the specification provided by the user and the metadata, even though when there is an implicit correspondence.

Considering that the discovery of adequate resources is a crucial task in the research activity in Life Sciences, it must be improved in order to make it easier for the researchers.

Chapter 3

Semantic Discovery of Web Resources in the Life Sciences

This chapter proposes a semi-automatic discovery approach that overcomes the main limitations presented by current approaches. The main goal of this discovery approach is to assist researchers in the discovery of the web resources that best fulfill their information needs, from the initial requirements specification until the final selection of the most appropriate resources, allowing the customization of the requirements and results.

First, in Section 3.1, we describe how the user's requirements are specified. Then, in Section 3.2, we present the main characteristics of the discovery process and, in Section 3.3, we give some conclusions about the proposed approach.

3.1 User's Requirements Specification

The requirements specification must provide a precise description of the user's information needs. In our approach, the requirements specification describes the tasks that have to be performed to achieve the user's goals as well as additional features the required resource must have, from now on *facets*, such as the input/output data types, the species involved in the resource and the method performed.

As discussed in Section 2.2.4, in current registries users are limited when describing their information needs and, sometimes, they do not get the expected resources due to a vague requirements specification. There are different types of users depending

on the way they express their information needs. For example, there are researchers who prefer describing their information needs with textual descriptions, whereas others prefer browsing through categories or more formal semantics.

Currently, different techniques to describe the users' requirements have been proposed to account for this diversity of users. Here, we propose a list of them that can be supported in our approach:

1. **Keywords-based queries,** consisting of a set of words, usually very specific, selected by the user to represent her information needs.

2. **Textual descriptions**. A textual description is an expression written in natural language in which the user describes her information needs, without any restriction of size nor vocabulary.

3. **Navigational search**. Registries with navigational search provide a hierarchy of categories describing different aspects of the resources. The user selects the category or categories that best describe her requirements and, then, all resources annotated with those categories are retrieved. Afterwards, the user has to manually analyze the retrieved resources and, if there is not any resource that fulfills her requirements, she has to refine the search by selecting other categories.

4. **Filtered search**. Filtered search is based on well-structured resource metadata. The registry provides filters on specific metadata whose values come from controlled vocabularies. To specify her information needs, the user has to select a value for each filter relevant to her requirement. Then, all resources that have that value in the specific field of metadata are retrieved.

5. **SPARQL**[1]. SPARQL Protocol and RDF Query Language is a syntactically-SQL-like language for querying RDF graphs via pattern matching. The language features include basic conjunctive patterns, value filters, optional patterns, and pattern disjunction. To use SPARQL as language of requirements specification, resources metadata must be represented in RDF. For example, using SADI resources, to retrieve all the names for UniProt *"P15923"*, we can use the following SPARQL query:

```
PREFIX sadi: <http://sadiframework.org/ontologies/properties.owl#>
```

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

```
PREFIX ss: <http://semanticscience.org/resource/>
PREFIX uniprot: <http://lsrn.org/UniProt:>
SELECT ?nameString
WHERE { uniprot:P15923 sadi:hasName ?name .
        ?name ss:SIO_000300 ?nameString . }
```

6. **Graphs**. They have been largely used as technique to represent data and, therefore, they have also been used to represent users requirements. A graph consists of nodes, which represent entities, and edges, which represent relationships between those entities. In the specification of the user's requirements, the nodes can represent the task to be performed or the values of the resources properties. Then, the graph is matched to the semantic representation of the web resources. SSWAP and SADI allow users to describe their requirements with graphs. Then, each graph is matched to the semantic representation of the web resources. Figure 3.1 shows a SSWAP RQG (Resource Query Graph), extracted from [36], that describes the query *Retrieve all resources that map anything to a taxa:Taxa*.

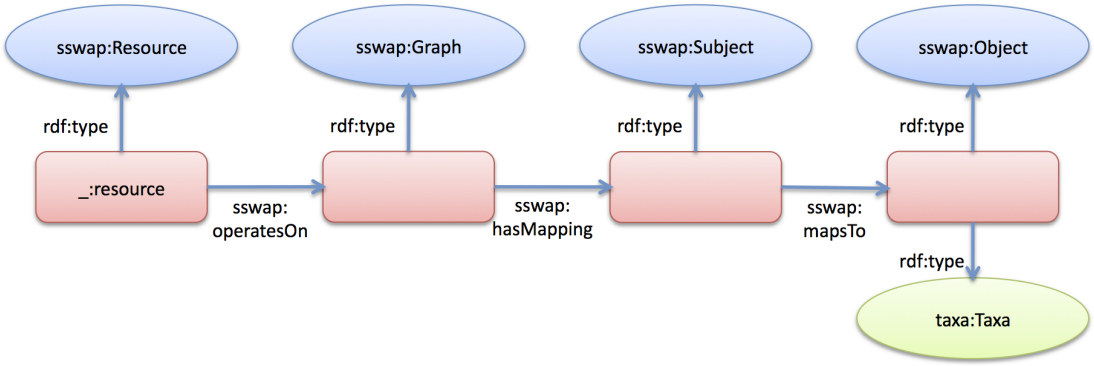

Figure 3.1: Requirements specification using a RQG

7. i^* **framework**. i^* formalism [115; 116], which is a goal-oriented and agent-oriented language, allows users to express their requirements by means of goals and tasks in a formal specification without taking into account the characteristics of the system. The i^* Strategic Rationale (SR) model describes user's interests and concerns and how they might be addressed. They are represented by means of *goal* and *task* elements described in natural language. For example, Figure 3.2 shows the i^* model that describes the user's information needs which consist

in comparing specific genes in different organisms. This information need is the user's goal, and it is fulfilled by executing a set of tasks: *retrieve protein sequences*, *predict gene structure* and so on. These tasks have to be performed by resources and, therefore, their descriptions are the input of the discovery process.

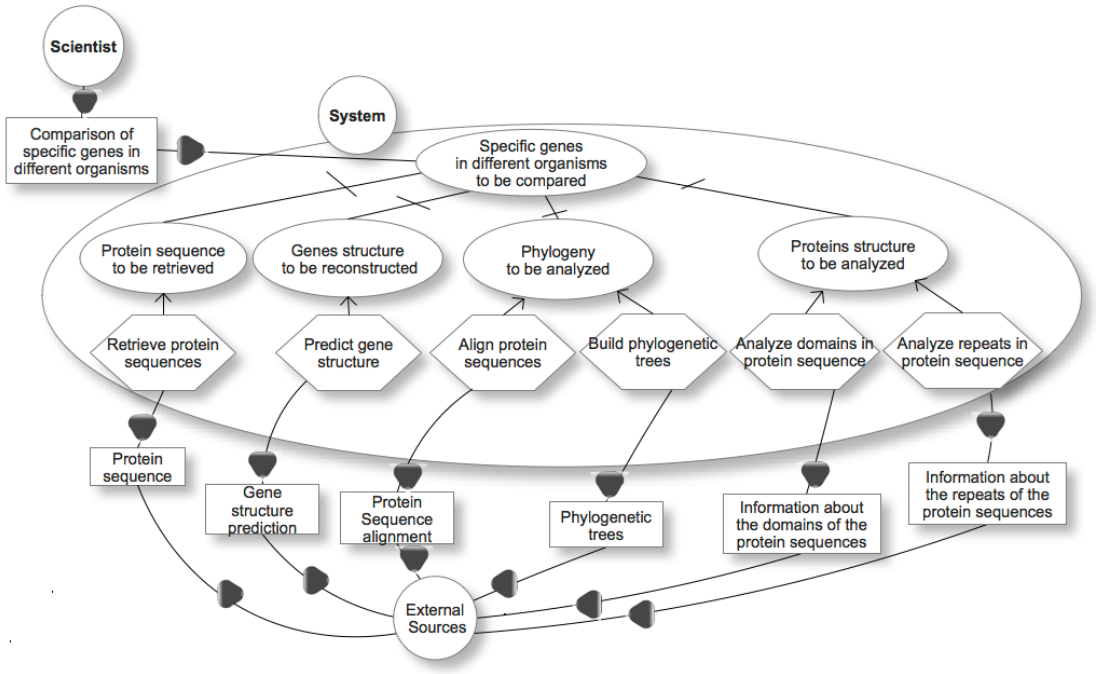

Figure 3.2: Requirements specification using the i^* model

Specific languages such SPARQL or i^* allow users to formally provide more information about their requirements that can be used to improve the discovery process, e.g., dependencies between resources can be extracted from the relations between elements in an i^* model. However, it supposes a huge cost for researchers, who are not experts on these technologies, to learn how to express their needs using these specific languages. In fact, there are already approaches, like BioQueries [34], that aim to help users to exploit the full potential of SPARQL to query biomedical databases. Unlike these languages, textual queries are the easiest and most intuitive way for users to describe their information needs, since they do not have to choose a limited set of keywords conforming to a specific vocabulary nor any specific query language.

To be widely adopted by users, the proposed approach is not restricted to a single requirements specification technique, but rather it can provide several techniques in order

to address different types of users. In order to make the discovery process independent from the technique used to specify the requirements, for each supported technique an extraction module must be defined to extract the information about the user requirement, as shown in Figure 3.3. In the most precise techniques such as SPARQL or i^*, the extraction module extracts information from the relationships and properties defined in the specification, e.g., the properties mentioned in a SPARQL query can define the value of a facet, and the dependency between two tasks in an i^* model relates the output of a task with the input of the subsequent task. The canonical model used for integrating these representations is introduced in the following sections.

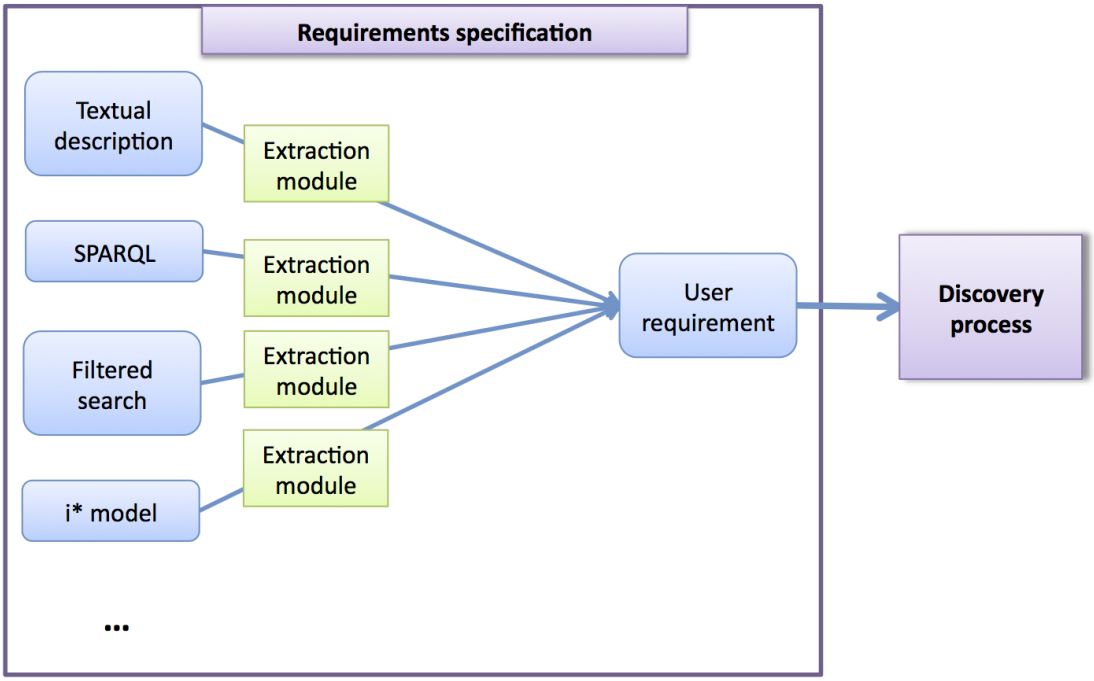

Figure 3.3: Architecture of the requirements specification module.

3.2 Semantic Web Resource Discovery

In this section, we describe the main characteristics of the web resource discovery approach proposed in this thesis, which assists the user in the discovery and selection of the most suitable resources for her information needs. The discovery is driven by a (rich) user's requirements specification and it is based on the normalization of data.

It consists of two phases, as depicted in Figure 3.4: *(i)* data normalization and *(ii)* discovery and ranking of resources.

Next sections describe the main characteristics of these two phases, which are further described in Chapter 4 and Chapter 5.

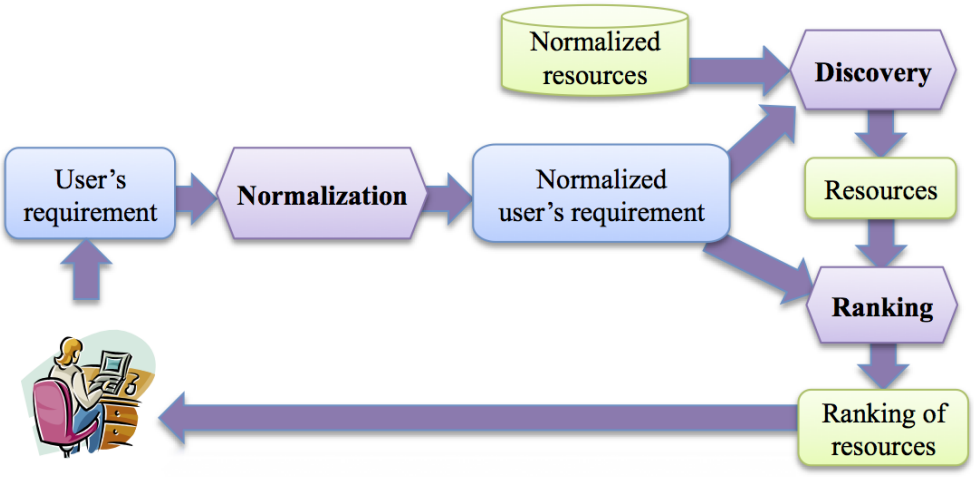

Figure 3.4: Overview of the proposed discovery process

3.2.1 Data Normalization

All data involved in the discovery process are semantically normalized. The normalization process represents the data (both user's requirements specification and resources metadata) in a machine-readable format and automatically identifies relevant information. It consists of two phases: *(i)* semantic annotation and *(ii)* knowledge extraction.

Firstly, the data is semantically annotated with widely accepted knowledge resources that formally describe the language used in Life Sciences web resource registries. The semantic annotation abstracts words to concepts well-described in external knowledge resources, reducing the heterogeneity and the ambiguity present in data and, more specifically, in textual descriptions. In contrast to other approaches, e.g., BioMoby and SADI, the semantic annotation is completely automatic, and it can annotate huge quantities of text with a minimal cost, which would be unfeasible with manual curation.

The final step of the normalization process is the automatic identification of relevant information implicitly described in the metadata. Textual descriptions usually contain

information about specific features of the resources which improve their characterization and which are relevant to the users. In this work, we refer to these features as *facets*.

As a result of the normalization, the data are enriched with formal knowledge, which alleviates the discovery dependency on data characteristics such the use of specific vocabularies or the lack of adequate metadata.

The normalization process is applied to the all data involved in the discovery process, i.e., the resources metadata and the user's requirements specification.

3.2.2 Discovery and Ranking

The discovery process is based on the semantic mapping between the normalized web resources metadata and the normalized user's requirements specification. The use of semantics allows retrieving resources described with different vocabularies but referring to the same concepts. Therefore, as stated in the thesis hypothesis, the normalization of data allows us to reconcile the user's information needs with the available web resources.

Finally, in order to assist the user in the selection of the most suitable resources, discovered resources are ranked according to their relevance to the user's requirement. The relevance is estimated on base to the similarity between the requirements specification and the resource characterization that also considers the accomplishment of facets. Additionally, each resource in the ranked list is attached with a summary of all its available metadata to help the user in the selection of the most suitable resource.

At the end, if the discovered resources are not those expected by the user, she can modify the initial requirement specification, the automatic identified facets values, or even discovery parameters, so that alternative resources can be explored.

3.3 Conclusions

In this chapter we have proposed a discovery approach for assisting users in the discovery of the most suitable resources for their requirements, addressing the main limitations of current registries.

Regarding the user's requirements specification, the proposed approach allows the user to provide a rich specification of her information needs describing the functionality and relevant features of the required resource. With the aim of being adopted by different types of users, the proposed approach is not restricted to a unique requirements specification format.

To alleviate the dependency of the discovery on the characteristics of the data, all the data involved in the discovery process, i.e., the requirements specification and the web resources metadata, are normalized. First, the data are semantically annotated and, then, relevant information is automatically identified by using knowledge extraction techniques.

Finally, the discovery of web resources is based on the semantic mapping between the normalized requirement specification and the normalized resources metadata. The use of semantics relaxes the mapping between resources and requirements, and allows retrieving resources described with different styles and vocabularies. At the end, unlike most current registries, the user is provided with a ranked list of resources in which the most relevant resources to the user's requirements are top-ranked. The resources are ranked on base to the fulfillment of the functionality and the features required by the user.

In conclusion, the discovery approach overcomes the main limitations presented by current registries thanks to: *(i)* a rich user's requirements specification not restricted by vocabularies nor formats, *(ii)* the use of semantics and knowledge extraction techniques to alleviate heterogeneity, ambiguity and implicitness issues, *(iii)* retrieval and ranking of resources driven by the functional task and the set of user-defined facets described in the requirements specification.

Chapter 4

Normalization

Data in Life Sciences are highly heterogeneous due to the lack of widely accepted standards. Moreover, the common use of natural language in the resources metadata and in the specification of the users' requirements makes even more difficult their automatic processing. Therefore, to not lose the expressivity of users but also to not limit the automatic processing carried out by computers, normalization techniques are required to represent the natural language information in a format that computers can understand and process.

Normalization is a process by which a textual description is transformed into a machine-processable representation. Some of the most simple and common normalization techniques are: removing punctuation, expanding abbreviations, converting all letters to lower and upper case, removing stopwords or word normalization. However, these techniques are too simple and do not represent the information properly.

In this chapter, we propose a normalization process, shown in Figure 4.1, that consists of two phases: *(i)* semantic annotation and *(ii)* knowledge extraction. Firstly, the text to be normalized is semantically annotated with knowledge resources in order to represent the data in a machine-processable format. In the second phase, relevant information about the resource described implicitly in the metadata is automatically extracted with knowledge extraction (KE) techniques.

First, in Section 4.1, we make a brief description of the knowledge resources used in the normalization process. Afterwards, Section 4.2 describes the proposed normalization process based on semantic annotations, and Section 4.3 describes the knowledge extraction method. Finally, Section 4.4 explains the characteristics of the normalization of the data involved in the discovery process, and Section 4.5 gives some conclusions

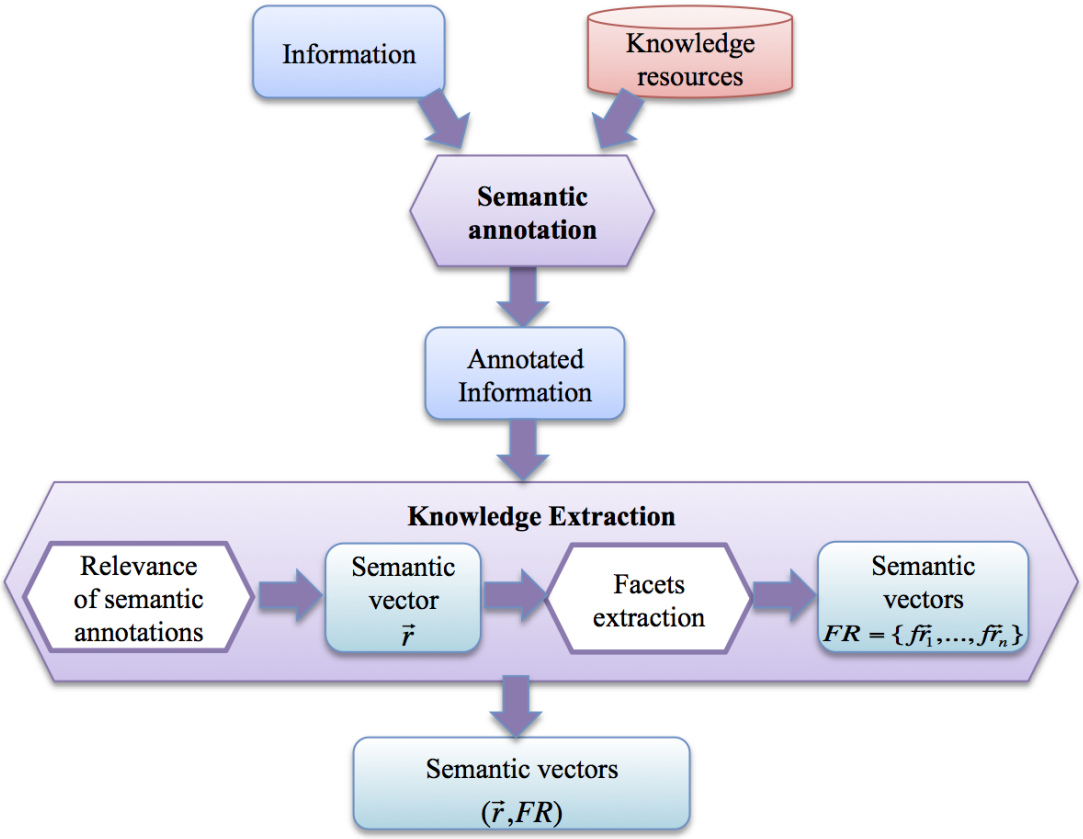

Figure 4.1: Overview of the normalization process

about the proposed normalization process.

4.1 Knowledge Resources Formalization

In this section we formalize the concept of knowledge resource (KR) and which are the minimal elements it must provide in order to be useful in semantic annotation and resource discovery.

Definition 4.1.1. *A knowledge resource (KR) is a formalization of the semantics of a domain by means of a set of concepts* $\mathcal{C} = \{c_1, ..., c_n\}$.

A concept $c \in \mathcal{C}$ *represents the semantic definition of a meaningful entity in a specific domain. A concept c consists of a semantic description of an entity and a set of lexical strings that represent the concept. The function* $lex : \mathcal{C} \rightarrow 2^{strings}$ *returns the lexical strings associated to a concept.*

Two concepts $c, c' \in \mathcal{C}$ *can be taxonomical related by either subsumption (is-a) or by 'broader-than' relationships. The taxonomical relationship between two concepts c and* c' *is represented as* $c \preceq c'$. *Let be* $ancestors(c) = \{c' \in \mathcal{C} | c \preceq c'\}$.

Usually, the domain covered by a KR is divided into a set of subdomains that have specific characteristics. Then, the concepts in the KR can be classified according to these specific subdomains using semantic types[1].

Definition 4.1.2. *A semantic type st represents a subdomain described in a KR. Let* $\mathcal{ST} = \{st_1, ..., st_m\}$ *be the set of semantic types describing the domain in a KR. Semantic types can be partially ordered by the subsumption relationships denoted as* $st_i \preceq st_j$.

A concept $c \in \mathcal{C}$ *has associated a set of semantic types. The function semtype :* $\mathcal{C} \rightarrow 2^{\mathcal{T}}$ *returns the semantic types of a concept.*

Knowledge resources can also be related on base to their level of specification in the description of a target domain.

Definition 4.1.3. *Let* $KR_i \preceq KR_j$ *be the relationship 'more specific than', in which* KR_i *provides a more specific representation of a domain than* KR_j.

[1] http://www.nlm.nih.gov/research/umls/META3_current_semantic_types.html

4.1.1 Knowledge Resources in Life Sciences

In Life Sciences registries, web resources descriptions mix terminologies of Computer Science, Biomedicine and Bioinformatics. Unfortunately, although there are efforts to build a unique vocabulary for Bioinformatics, e.g., [2], a single comprehensive ontology covering all these terminologies does not exist yet. Therefore, several existing knowledge resources need to be combined to cover the domain vocabulary. For this purpose, we have selected several ontologies that cover different terminologies of this domain, namely:

- **UMLS Meta-thesaurus (version 2010AA)**[1] covers concepts about procedures, anatomy, diseases, proteomics and genomics. This metathesaurus is an integrated resource that includes a great variety of thesauri and ontologies such as the Gene Ontology (GO)[2], the HUGO database[3], and many other related to the biomedical domain.

- **EDAM** [86] (EMBRACE Data and Methods) ontology includes concepts strictly in the domain of bioinformatics such as bioinformatics operations, topics, types of data and formats. General computer science or biological concepts are not included in the ontology.

- **myGrid ontology**[4] is the reference ontology of myGrid project (e.g., BioCatalogue and myExperiment), and it has been developed for semantic service discovery. It is divided in two distinct components: the *service ontology*, which describes physical and operational features of resources such as input and output data types, and the *domain ontology*, which describes core bioinformatics data types and their relationships to one another.

- **Bioinformatics in Wikipedia**. In order to provide broad coverage for the names of the algorithms and methods involved in bioinformatics, we have included the entries of the Wikipedia related to any subcategory of the Bioinformatics category. A tailored lexicon with Wikipedia entries related to Bioinformatics has been built using the tool presented in [55], which automatically builds tailored lexicons for domains that are not well covered by rich KRs.

[1] http://www.nlm.nih.gov/research/umls/
[2] http://www.geneontology.org
[3] http:/www.genenames.org
[4] http://www.mygrid.org.uk/tools/service-management/mygrid-ontology

- **Named entities**. This lexicon covers named entities that are not described in the other KRs and that are relevant in our corpus. It includes the name of popular formats (e.g., PDF, PS, and so on), names of algorithms (e.g., Smith & Waterman, Myers and Millers), names of popular resources (e.g., Clustalw, MUSCLE), acronyms (e.g., mol, seq) and other relevant named entities.

These knowledge resources are partially related to each other on base to their level of specificity as follows:

$$Named\ Entities \preceq myGrid \preceq EDAM \preceq UMLS$$

$$Bioinformatics\ in\ Wikipedia \preceq UMLS$$

For tagging purposes, all these KRs are loosely integrated into a concept repository, i.e., a lexicon, which consists of an inventory of concepts, their taxonomical relationships (i.e., $\preceq$ relationship), and the lexical variants associated to each concept (e.g., alternative labels, synonyms, and so on) [47]. In order to provide a common representation of the integrated KRs, all the concepts are represented with the following notation:

$$KRreference ::= KR : concept(: ST)?$$

Table 4.1 shows the concept representation of each one of the KR described above. This notation is inspired in the competitions CALBC for annotating large corpora.

KR	**Concept reference format**	**Comment**
UMLS	`UMLS:C<number>:STypes`	`STypes` are the semantic types associated to UMLS concepts (e.g. Disease, Protein, etc.)
EDAM	`EDAM:E<number>`	Concepts extracted from the EDAM ontology.
myGRID	`myGR:D<number>`	Concepts extracted from the ^{my}Grid ontologies.
Wikipedia	`Wiki:W<number>:Categs`	`Categs` are the categories associated to the page entry of the referred concept.
Named Entities	`OTHR:KR<number>:STypes`	`STypes` are the semantic types associated to UMLS concepts

Table 4.1: Concept reference formats used for the different knowledge resources.

4.2 Semantic Annotation

The semantic annotation (SA) is the process of linking the *entities* in a text to their *semantic descriptions*, which are stored in KRs such as thesauri and domain ontologies.

In this thesis, we have adopted an automatic and unsupervised annotation method [13]. This method is based on concept retrieval, that is, it finds the most relevant concepts w.r.t. the text words and, then, selects those that best cover the underlying text semantics. This annotation tool was tested within CALBC competition over a collection of 150.000 PubMed abstracts about immunology [92] using UMLS as KR. We have chosen this tool against other annotation tools, like MetaMap or BioPortal, because of its easy parametrization, its high recall, and the possibility of including several lexicons. In this thesis, the semantic annotator considers several KRs simultaneously in order to cover the different terminologies used in Life Sciences.

Using this annotator, the semantic annotation of a textual description consists of a set of concepts, defined in a KR, linked to the words in the text that represent entities described by these concepts.

Definition 4.2.1. *A semantic annotation of an entity E is a pair* $<E,\{c_i\}>$*, where* $c_i \in \mathcal{C}$ *are the concepts that semantically describe E.*

The annotation process is divided into three phases: *(i)* selection of target text chunks that likely contain an entity, *(ii)* concept retrieval through the mapping between each text chunk and the lexical variants of each KR concept in order to retrieve the list of concepts that are potentially associated, and *(iii)* a post-processing of the resulting annotation to make it more accurate and precise. Next, each one of these phases are described with more details.

4.2.1 Target Text Chunks Selection

A textual description consists of a set of meaningless words, such as prepositions or conjunctions, that do not provide relevant information and a set of words that provides the information described in the text. The meaningful words represent entities that are potentially contained in a KR. In the literature, there are several works about Biomedical Named Entity Recognition (NER) using natural language processing (NLP) [87; 117]. However, they are focused only on specific biomedical entities such as protein, DNA, and so on. In this work, in order to identify all the meaningful entities described in a textual description, this is splitted up into chunks.

Definition 4.2.2. *A chunk is a minimum text segment that likely contain an entity that is formally described in a knowledge resource.*

Entities are usually represented by noun phrases; thus the chunks to be considered as relevant are restricted to be noun phrases. There are several methods to extract chunks from a textual description, from ad-hoc regular expressions representing frequent chunks patterns to NLP tools such as OpenNLP[1] or GeniaTagger[2]. The selected semantic annotator is not affected by the size or the characteristics of the extracted chunks whenever it respects the boundaries of the intended entities.

4.2.2 Concept Retrieval

The objective of the semantic annotation is to select the set of concepts that best describes a text chunk. The best concepts are those that are less ambiguous, more compact and match the maximum number of words of the text chunk.

Definition 4.2.3. *Given a text chunk T, let $candidates(T) = \{c \in \mathcal{C} | \exists w \in T \wedge \exists s \in lex(c) \wedge w \in s\}$ be the set of concepts that match one or several words in the text chunk T and, therefore, are candidates to describe T. Let $matched\_words(c) = \{w \in T | \exists s \in lex(c) \wedge w \in s\}$ be the set of words in T covered by the concept c.*

Each concept $c_i \in candidates(T)$ is evaluated according to an information-theoretic function, which is inspired by the matching function defined in [66] and the word content evidence defined in [25].

Definition 4.2.4. *Let $sim(c, T)$ be the function that measures the information coverage of T with respect to each lexical variant of the concept c. It is calculated by:*

$$sim(c, T) = max_{S \in lex(c)} \left[\frac{info(S \cap T) - info(S - T)}{info(S)} \right] \tag{4.1}$$

Information is measured with an estimation of the string words entropy in a background corpus $\mathcal{G}$:

$$info(S) = -\sum_{w \in S} log(P(w|\mathcal{G})) \tag{4.2}$$

[1] http://opennlp.apache.org/
[2] http://www.nactem.ac.uk/GENIA/tagger/

The relevance of a word is measured by means of its estimated probability within a background corpus $\mathcal{G}$. In this way, highly frequent terms in the background corpus contribute little to the final score of the strings containing them. In this work we use the whole Wikipedia (snapshot 2008) as background corpus $\mathcal{G}$.

All concepts with the same score and matched words are grouped together. A minimum threshold is defined over the score of concepts in order to reduce the number of concept groups to seek and evaluate. Then, each group with a score greater than the threshold is evaluated according to the following criteria: the ambiguity of the group (i.e., number of different concepts), the maximum gap between the matched words in the chunk and the number of matched words. The concepts less ambiguous, more compact and with larger matches are top-ranked. A more detailed description of the annotation process is provided in [13].

This process is executed in parallel for each one of the KRs and, as a result, a ranked list of concepts $LC(T)$ of each KR is obtained for the text chunk T.

Finally, given the ranked lists of concepts $LC(T)$ for the text chunk T, the parts of T associated to each concept have to be identified. The tagging process is based on the Algorithm 1. As result of this tagging process, the semantic annotation of a chunk T is a set of concepts of different KRs that describe the entity in T.

Algorithm 1 Text Tagger

Require: A text T and the ranked retrieved concepts $LC(T)$
Ensure: The text tagged with a covering of concepts form $LC(T)$.
  Initialize $CoveredW = \phi$
  while $CoveredW \neq T$ and $LC(T) \neq \phi$ **do**
    pop c_i from $LC(T)$
    record the positions of c_i's words in T
    append to $CoveredW$ the c_i's matched words
  end while

Example 4.2.1. Given the chunk *nucleotides*, its semantic annotation is:
<nucleotides, {E0001207.8, E0000022.8, C0028630.8}>.

Definition 4.2.5. *Given the semantic annotation of the chunk T, the function annotation(w) returns the set of concepts covering the word w in T.*

Example 4.2.2. In the last example, the function *annotation*(*nucleotides*) returns {E0001207, E0000022, C0028630}.

Annotations can be ambiguous when there is more than one concept from the same KR with different semantic types referring to the same entity.

Definition 4.2.6. *A semantic annotation a is ambiguous when there are at least two concepts $c, c' \in a$ from the same KR such that $matched\_words(c) = matched\_words(c')$ and $semtype(c) \neq semtype(c')$.*

Finally, the semantic annotation of a textual description is represented by a semantic vector containing all concepts in the annotation.

Definition 4.2.7. *Let $\overrightarrow{d} = \{c_1 : p_1, ..., c_n : p_n\}$ be the semantic vector representation of a text description d in which c_i is a concept in the annotation of d, and p_i is its associated weight defined by its $tf*idf$ value, where $tf(c)$ is the frequency of the concept c in the description and $idf(c)$ is defined as follows:*

$$idf(c) = max_{S \in lex(c)} info(S) \tag{4.3}$$

With $concepts(\overrightarrow{d})$ we denote the set of concepts that appear in the semantic vector $\overrightarrow{d}$.

Example 4.2.3. Given the textual description *"BLAST finds regions of similarity"*, its semantic annotation is:
{<BLAST, {C0523113.8, W363695.10, E0000646.6}>, <regions, {C0017446.5, C1514562.5}>, <similarity,{C2348205.8}>}.
Its associated semantic vector is:
$\overrightarrow{d}$ = {*C0523113:8, W363695:10, E0000646:6, C0017446:5, C1514562:5, C2348205:8*}
The function $concepts(\overrightarrow{d})$ returns: {C0523113, W363695, E0000646, C0017446, C1514562, C2348205}.

Complexity. The estimation of the information coverage of the lexical strings (Formula 4.2) has a cost of $O(W)$, being W the number of words in the vocabulary. It is calculated only once, thus it does not affect the semantic annotation process.

The cost of the semantic annotation of a textual description is proportional to $O(N_c)$, being N_c the number of concepts in the KR. However, this cost is usually lower since the KR lexicons are stored in inverted files, which make the retrieval of the candidate concepts more efficient.

4.2.3 Semantic Annotation Post-Processing

The use of several KRs in the mapping function produces annotations with a high number of concepts which usually differ on their specificity degree. Moreover, many of these concepts introduce ambiguity to these annotations. Ambiguous concepts usually come from broad KRs, in which the definitions of some concepts are not precise and, in consequence, the degree of specificity of these concepts is usually very low. Therefore, in a semantic annotation with multiple concepts, the concepts with a higher degree of specificity are assumed to be the ones that best describe the matched entity. Thus, in order to obtain more precise annotations, only the most specific concepts must be kept to describe the matched entity.

The specificity of a concept can be estimated by its *idf* score (Formula 4.3). General concepts appear more frequently than specific concepts and, therefore, a general concept has a lower *idf* value than a specific concept. So, concepts with very low *idf* scores can be candidates to be removed from the original annotation.

In this thesis, we propose two post-processing techniques to simplify the semantic annotations that take into account the *idf* score of the concepts: *(i)* simplification of multi-word entities annotations, and *(ii)* simplification of multiple annotations of single word entities. Next, each proposed simplification technique is described.

4.2.3.1 Simplification of Multi-Word Entities Annotations

The semantic annotator is able to annotate multi-word entities, like *"multiple alignments"*. In this kind of annotations, apart from the concepts matching all words, the annotator may also select concepts that match subsets of words of the text chunk. However, the relevant meaning of an annotation is usually represented by the longest matching concepts, which are more specific than those matching subsets of words. Moreover, as frequent concepts subsumed by more specific ones are meaningless and usually introduce noise in the annotations, the concepts annotating subsets of words must not be kept in the annotation.

The first simplification consists in selecting the concepts with non-overlapped longest matches covering the whole annotation. Moreover, from the concepts with the same match, those whose *idf* score is lower than a threshold are rejected.

Example 4.2.4. The semantic annotation of the chunk *protein domain* is:
{<protein, {E0000065.10, E0001468.10}>, <domain, {E0000065.10, E0001468.10, C1883221.4,

C1883204.4, D9000300.15, C1514562.4, C9000023.6}>
As it can be noticed, apart from the concepts matching both words, there are concepts matching the single word *domain*. In the simplification, the concepts matching single words are rejected, and only those matching both words remain (E0000065, E0001468). In this case, the two concepts matching the two words remain because they have an *idf* score (Formula 4.3) higher than a threshold (set to 8 in these examples). Then, the simplified annotation is:
{<protein, {E0000065.10, E0001468.10}>, <domain,{E0000065.10, E0001468.10}>}.

However, in case that there is a word in the annotated chunk that is not matched by any of the selected concepts, all the concepts with an *idf* score higher than a threshold are kept in the annotation in order to not lose information.

Example 4.2.5. The semantic annotation of the chunk *nucleotide query sequence* is:
{<nucleotide, {E0001207.8, E0000022.8, C0004793.13, D9000378.13}>, <query, {C1522634.8}>, <sequence, {C0004793.13, D9000378.13, E0000080.5, E0002044.5}>}
As it can be noticed, there are concepts matching *nucleotide sequence* (C0004793, C0004793, D9000378), but there is no concept matching *query sequence* or *nucleotide query* or *nucleotide query sequence*. So, if we only select the concepts with longest matches, the word *query* is not covered. Therefore, the concepts matching single words are not rejected.

Finally, only the concepts with an *idf* score lower than 8 are rejected. In this case, the concepts E0000080 and E0002044 (sequence) are removed from the original annotation since their *idf* score is 5.
{<nucleotide, {E0001207.8, E0000022.8, C0004793.13, D9000378.13}>, <query, {C1522634.8}>, <sequence, {C0004793.13, D9000378.13}>}.

One additional effect of this post-processing technique is to smooth the frequency of some single-word concepts that can bias the search in the discovery process.

4.2.3.2 Simplification of Multiple Annotations of Single Word Entities

A single word chunk can be annotated with one or more concepts from a single KR or from several KRs. In the annotations with multiple concepts, the concepts usually represent different degrees of specificity. There are concepts that are very ambiguous in the broader KRs as they cover more topics. Specific KRs are focused on the collection at hand and, therefore, can provide a more appropriate semantics to the identified

entities. So, in order to provide a more precise representation, the annotation can be simplified keeping only the most specific concepts of the most specific KRs.

Therefore, to determine the specificity of a concept, apart from its *idf* score, we consider the specificity of its KR and the relationship between concepts in the KRs.

So, the simplification of the annotation of a single word entity with multiple concepts is divided into three phases:

1. **Selection of the concepts with a high *idf* score**. All the concepts whose *idf* score is higher than a threshold remain in the semantic annotation because they are consider specific concepts.

2. **Selection of the most specific KRs.** From those concepts whose *idf* value is lower than a threshold, if they come from different KRs, the most specific are those from the most specific KR. The most specific KR is determined on base to the KRs relationship $KR_i \preceq KR_j$.

3. **Selection of the most specific concepts.** Finally, from those concepts of the most specific KR, the most specific concepts among them are selected on base to the concepts taxonomical relationship $c_i \preceq c_j$.

So, after executing these three steps, the concepts with an *idf* score higher than a threshold and the most specific concepts from the most specific KRs remain in the semantic annotation.

Example 4.2.6. The word *sequence* is annotated with concepts from UMLS, (C1547787, C1610719, C0004793), and with concepts from EDAM ontology, (E0000080, E0002044). In this case, the UMLS concepts are rejected because their *idf* scores are lower than 8, and because there are concepts from a more specific KR, $EDAM \preceq UMLS$. Then, from the two EDAM concepts, E0000080 is selected since E0000080 $\preceq$ E0002044. Therefore, the resulting annotation is: <sequence, {E0000080.5}>.

4.3 Knowledge Extraction

In current open registries search engines, all words in a textual description are treated in the same way independently of their semantics and relevance. In a textual description, there are words more relevant than others since they represent the relevant information in the text. Moreover, there are words in the textual descriptions that implicitly

represent some aspects of the described resource. Neither relevance nor semantics are taken into account by current open registries since most of them only rely on string matching techniques. However, this information can improve considerably the characterization of a resource and, consequently, related tasks such as their discovery and the interoperability.

In order to extract this relevant information, we use knowledge extraction techniques. First, the relevance of each concept in a semantically annotated description is estimated using a topic-based model. Then, the concepts that may describe features are identified using semantic and probabilistic techniques.

4.3.1 Resource Characterization

In a textual description not all words have the same relevance due to their semantics and the context in which they appear. For example, in *"define structurally and functionally important domains of the membrane", "predict gene functions" and "compare functional relationships"*, the concept *function* does not have the same relevance. In the first sentence, *functionally* describes only a characteristic of the domain, in the second one, *function* is the key concept in the query, since it is the object that must be predicted and, finally in the third one, *functional* specifies the type of relationship that must be compared. Therefore, the relevance of a word in a specific text needs to be estimated.

Language modeling techniques consider each document as a language model that determines the probability of emitting each word from the document. This probability is calculated by maximum likelihood estimation (MLE) as follows:

$$p(w|d) = \frac{n(w,d)}{\sum_{w_i \in d} n(w_i,d)} \tag{4.4}$$

where $n(w,d)$ returns the number of times the word w appears in the document d.

For retrieval tasks, this probability should be smoothed, so that non-zero probabilities can be assigned to query terms that do not appear in a given document. One of the simplest smoothing method consists in using a linear interpolation (Jelinek-Mercer [46]) with a background collection model $p(w, \mathcal{G})$:

$$p_\lambda(w|d) = \lambda \cdot p(w|d) + (1-\lambda) \cdot p(w|\mathcal{G}) \tag{4.5}$$

[63] views this smoothed model as coming from a simple 2-state hidden Markov

model, and trains the parameter λ using MLE. However, this formula does not consider the word as a part of a bag of words, and do not take into account the context in which the word appears. [56] proposes a smoothed version that takes into account the context of the document to create a tailored language model.

However, the relevance of a word in a text depends on the context in which it appears and on its semantics. In the description of web resources, the context is related to the resource functionality, so the relevance of a word will depend on the described functionality.

Text categorization [98] aims to classify documents according to their content and characteristics. Traditional text categorization relies on either classification rules manually defined by domain experts or large sets of labeled examples, which in some domains are difficult to find. Currently, there are approaches such as [42; 61] that use unlabeled examples to improve the categorization. Nevertheless, text categorization is not sufficient to characterize resources, since web resources metadata usually describe a mixture of topics, each one with a different distribution.

Topic-based models [101] are able to identify the topic or the mixture of topics a text is about, by considering the distribution of words in a text. These models consider topics as distributions of words and documents as a distributions of topics. In our scenario, resource metadata mainly describe the functionality, that can be seen as a topic since in Life Sciences, the resources functionality is mostly classified [103]. Therefore, topic-based models can be used to identify the functionality described in the resource metadata and, therefore, calculate the relevance of each concept with respect to this functionality.

LDA (Latent Dirichlet Allocation) [17; 40] is a statistical model of document collections that aims to discover the topics that are hidden in the documents. LDA is mostly described by its generative process by which the model assumes the documents arose. Each document exhibits the topics in different proportion, and each word in each document is drawn from one of the topics. However, it assumes that topics are hidden and they must be estimated. Unfortunately, LDA topics bias to frequent co-occurrences of terms and, consequently, topics are dominated by frequent tasks, e.g., sequence analysis tasks such as the alignment of sequences or the comparison.

In this thesis, we propose a topic-based model to identify the topics described in the web resources metadata, and to calculate the relevance of each concept in the web resource characterization. In our work, topics are about the biomedical tasks underlying

both web resources and user's requirements. Existing annotations of the resources, e.g., tags, allow us to automatically estimate the corresponding topics models. Next, the proposed topic-based model is formally described.

4.3.1.1 Topic-based Model

The purpose of the topic-based model is to calculate the probability of all concepts $c \in KR$ for each defined topic.

Definition 4.3.1. *Let $\mathcal{T} = \{t_k\}_{1 \leq k \leq n}$ be the set of base tasks, i.e., topics, represented in the textual descriptions of a domain. Let RT_k be a set of semantically annotated descriptions deemed relevant for the base task t_k. The probability of a concept $c \in KR$ for a base task t_k is estimated as:*

$$p(c|t_k) \propto \sum_{\overrightarrow{d}_j \in RT_k} p(c|\overrightarrow{d}_j) \cdot p(\overrightarrow{d}_j|t_k) \tag{4.6}$$

It is worth mentioning that, to calculate the distribution $p(c|\overrightarrow{d}_j)$, not only the concepts in $\overrightarrow{d}_j$ are considered, but also the common ancestors of these concepts in the KR.

Definition 4.3.2. *Let define $common\_ancestors(\overrightarrow{d}) = \{c \in \mathcal{C} | \exists c_1, c_2 \in concepts(\overrightarrow{d}) \wedge c_1 \preceq c \wedge c_2 \preceq c\}$ as the concepts that are the common ancestors in the KR of the concepts in $\overrightarrow{d}$.*

For each concept $c \in concepts(\overrightarrow{d}) \cup common\_ancestors(\overrightarrow{d})$, its probability is estimated from its frequency in $\overrightarrow{d}$, and it is smoothed by propagating it through its ancestors using random walks [99]. This smoothed probability is calculated as follows:

$$p(c|\overrightarrow{d}) \propto \sum_{c_i \in concepts(\overrightarrow{d})} p^*(c|c_i) \cdot p^1(c_i|\overrightarrow{d}) \tag{4.7}$$

where $p^*(c|c_i)$ is the weight of the edge between c and c_i in the random walk matrix, and $p^1(c_i|\overrightarrow{d})$ is estimated from the concept frequency in the description and smoothed with Jelinek-Mercer as follows:

$$p^1(c_i|\overrightarrow{d}) = \lambda \cdot \frac{tf(c_i, \overrightarrow{d})}{\sum_{c_k \in concepts(\overrightarrow{d})} tf(c_k, \overrightarrow{d})} + (1 - \lambda) \cdot p(c_i|\mathcal{G}) \tag{4.8}$$

where $p(c_i|\mathcal{G})$ is the probability of the concept c_i in a background corpus $\mathcal{G}$.

The second probability in Formula 4.6, $p(\overrightarrow{d}|t_k)$, represents the chance of retrieving $\overrightarrow{d}$ as a relevant description in the context of t_k. This probability is estimated by sampling instances of t_k and counting how many times each description $\overrightarrow{d}$ is discovered. Thus, the probability is calculated with MLE as follows:

$$p(\overrightarrow{d}|t_k) = \frac{n(\overrightarrow{d}, t_k)}{\sum_{\overrightarrow{d}_i \in RT_k} n(\overrightarrow{d}_i, t_k)} \tag{4.9}$$

where $n(\overrightarrow{d}, t_k)$ returns the number of times $\overrightarrow{d}$ is retrieved with tk's instances. We consider that an instance of a task is an example description of the base task.

Table 4.2 shows the top-5 ranked concepts for a set of Life Sciences topics that correspond to research base tasks.

Complexity. With this topic-based model, all the concepts $c \in KR$ have a different probability for each base task, being in this way more significant for some topics than for others. The creation of this topic-based model has a cost proportional to $O(N_c^2)$, and it requires $O(N_c \cdot N)$ storage, being N_c the number of concepts in the KR and N the number of textual descriptions. Although the cost of this topic-based model is high, it is created only once, and therefore it does not affect the discovery process.

4.3.2 Facets Extraction

Textual descriptions may contain information related to specific features of the item being described. These features are described implicitly, and the words describing them are hardly ever considered as feature values. For example, web services descriptions usually contain information about the input and output data types of the service. Therefore, techniques to extract relevant information about the resources features from textual descriptions are required. These features are called *facets* in IR systems.

Definition 4.3.3. *A facet represents a characteristic of a resource that is relevant for its characterization and retrieval. A facet takes the form of an attribute-value pair.*

Faceted search systems [94] enable the classification of the information in multiple dimensions corresponding to the different facets. This contrasts with traditional taxonomies in which the hierarchy of categories is fixed and unchanging, e.g., [39]. Most faceted search systems are based on well-defined metadata, usually represented

Topic	Top-5 concepts
Sequence similarity search	D9000518 (sequence similarity) E0001413 (similarity sequence) D9000419 (protein sequence) E0002976 (protein sequence) C2348205 (similarity)
Phylogeny	E0000080 (sequence) D9000370 (molecular structure) D9000400 (phylogenetic tree) E0000872 (tree) E0000191 (phylogeny)
Sequence alignment	E0000083 (alignment) C1706765 (alignment) E0000504 (multiple alignment) E0002976 (protein sequence) E0000159 (sequence comparison)
Find genes with functional relationships	E0000198 (pathway) E0000581 (database) E0001208 (protein) D9000418 (protein interaction record) M9000027 (location, interface)
Analyze transgenic model organism	E0000200 (microarrays) M9000027 (location, interface) C1709016 (microarray) E0000197 (gene regulation expression) D9000322 (gene report expression)
Proteins with a functional domain	C9000023 (domain) E0001208 (protein) E0000581 (database) E0000170 (motifs) E0002976 (protein sequence)
Predict structure	D9000423 (protein structure) D9000507 (secondary structure prediction) E0002814 (protein structure) E0001208 (protein) E0001213 (RNA)

Table 4.2: Top-5 ranked concepts for Life Sciences topics

as facet-values pairs. However, in open registries, facets are usually described in textual descriptions. In the literature, there are some approaches that identify facets from texts. [20] proposes a faceted topic retrieval system and compare LDA [17] and relevance modeling [58] as methods to automatically extract facets from documents. They consider facets as any information need, e.g., topics, while we consider facets as addi-

tional features of the resources that are independent of topics. Therefore, neither LDA nor relevance modeling can be applied in our approach as proposed in the literature. [27] proposes an unsupervised technique that fully automates the extraction of useful facets from free-text, expanding it with terms that appear in the context of the identified relevant terms in external resources such as Wikipedia. However, they assume that facets are mutually exclusive and, therefore, a term corresponds to only one facet. This for example cannot be applied to the facets "input" and "output" of web services.

Our facets extraction method extracts relevant information for each user-defined facet independently of the topics. We distinguish two types of facets: *(i)* facets whose values are identified by a specific semantic type, and *(ii)* facets whose values can be of different semantic types. Therefore, we use two different techniques to retrieve both types of facets: one based on semantics and another based on a probabilistic model. Both techniques calculate the probability of a concept representing a facet in a textual description, which is used to characterize the resource being described. Next, both facets extraction methods are further explained.

4.3.2.1 Semantic Facets Extraction

There are facets whose values are represented by concepts of specific semantic types in a KR. We refer to these facets as *semantic facets*.

Definition 4.3.4. *A semantic facet is a facet whose values can be identified by the semantic type of their associated concepts. A semantic facet sf is defined by a set of semantic types $\{st_j\}_{st_j \in \mathbb{ST}}$. The function $semtypefacet(f)$ returns the semantic types of the facet f.*

Example 4.3.1. Textual descriptions of web resources in Life Sciences may mention which species or which diseases the information provided by the resource is related to. Both can be considered as facets that are identified by semantic types. The values of the facet species conform to semantic types like Bacteria, Virus, and so on, whereas the values of the facet referred to the disease are associated to semantic types like Diseases and Syndromes.

So, given a semantically annotated text, the concepts that have the semantic types associated to a facet are automatically selected as candidate values for that facet.

Definition 4.3.5. *Let $candidates(sf)$ be the set of concepts whose semantic types are*

those associated to the semantic facet sf and, therefore, they are candidates to represent values of such a facet.

To calculate the probabilities of the candidate concepts, all available textual descriptions are considered. From now on, let us consider $\mathcal{D} = \{\overrightarrow{d}_1, ..., \overrightarrow{d}_n\}$ be the set of all available textual descriptions in the catalogue.

Definition 4.3.6. *Let define $\overrightarrow{d}[sf] = \{c : p|c : p \in \overrightarrow{d} \wedge c \in candidates(sf)\}$ as the semantic vector that contains the candidate concepts in $\overrightarrow{d}$ representing values of the semantic facet sf.*

The probability of a concept $c \in candidates(sf)$ describing a value of sf is calculated as follows:

$$p_{sf}(c) = \frac{p(c|sf) \cdot p(sf)}{p(c)} \tag{4.10}$$

The probability $p(c|sf)$ is estimated by its relative frequency as facet value in the whole catalogue:

$$p(c|sf) = \frac{\sum_{\overrightarrow{d}_s \in \mathcal{D}} tf(c, \overrightarrow{d}_s[sf])}{\sum_{\overrightarrow{d}_s \in \mathcal{D}} \sum_{c_k \in sf} tf(c_k, \overrightarrow{d}_s[sf])} \tag{4.11}$$

where $tf(c, \overrightarrow{d}[sf])$ returns the frequency of the concept c in the semantic vector $\overrightarrow{d}[sf]$.

Finally, the probability that the concept c represents the semantic facet sf in a textual description $\overrightarrow{d}$ is estimated as follows:

$$p(c, sf|\overrightarrow{d}) = p_{sf}(c) \cdot p(c|sf, \overrightarrow{d}) \tag{4.12}$$

$$p(c|sf, \overrightarrow{d}) = \frac{tf(c, \overrightarrow{d}[sf])}{\sum_{c_k \in \overrightarrow{d}[sf]} tf(c_k, \overrightarrow{d}[sf])} \tag{4.13}$$

Complexity. The estimation of the probabilities $p_{sf}(c)$ (Formula 4.10) and $p(c|sf)$ (Formula 4.11) of n_{sf} concepts (those having the semantic types associated to sf) has a cost $O(n_{sf} \cdot N)$ (being N the size of D) and requires $O(n_{sf})$ storage, since the number of the semantic facets is constant w.r.t. n_{sf}. Both $p_{sf}(c)$ and $p(c|sf)$ are calculated only once, and they are stored in inverted files. The cost of calculating the probability of a concept to represent a specific semantic facet in a specific textual description is

proportional to the number of concepts in the description with the semantic types $semtypesfacet(sf)$.

4.3.2.2 Probabilistic Facet Extraction

There are some facets whose values cannot be identified only by their semantic types, but also by the context in which the concept is expressed. We propose a probabilistic model to determine if a concept represents a value of a specific facet in a textual description based on its co-occurrence with facet keywords.

Definition 4.3.7. *A probabilistic facet f is defined by $K_f = \{c_1, ..., c_k\}$, a set of concepts associated to keywords defined by the user as relevant to identify f, and by $V_f = \{c_1, ..., c_n\}$, a set of concepts that are known to be values of facet f. Let define $\mathcal{F} = \{f_i\}_{1 \leq i \leq f}$ the set of user-defined probabilistic facets.*

Example 4.3.2. The words *results* and *returns* are keywords of the facet output and *implements* and *performs* are keywords of the facet method, while *report* is a known value of the facet output and *Smith-Waterman* is a known value of the facet method.

For each user-defined probabilistic facet, the user defines an initial set of keywords K_f. Then, the set V_f is built automatically using a translation model; which is also used to refine the initial set of keywords K_f.

Let $V = \{c_1, ..., c_n\}$ be the whole set of concepts used in the textual descriptions. The probability of a concept c to represent a value of the facet f can be calculated as:

$$p_f(c) = \sum_{c_j \in V} p(c|c_j) \cdot p^*(c_j|f) \tag{4.14}$$

where $p^*(c_j|f)$ represents the probability of c_j in the initial set of keywords of the facet f, which can be estimated as follows:

$$p^*(c_j|f) = \begin{cases} \frac{1}{|K_f|} & if \quad c_j \in K_f \\ 0 & otherwise \end{cases} \tag{4.15}$$

To estimate the entailment of concepts $p(c|c_j)_{c,c_j \in V}$, we rely on a translation model [52]. To build the translation model, all the available textual descriptions are previously divided into text segments of variable size. First, each textual description is pos-tagged and, then, text segments with the structure (VP, NP+) are extracted, where VP is a verb phrase and NP is a noun phrase.

Definition 4.3.8. *Let define a c-chunk as the set of concepts associated to the words in each extracted text segment with the syntactic structure (VP, NP+). The function f_chunks(d) returns all c-chunks associated to a textual description d. Let consider W as the set of all possible c-chunks of all the textual descriptions in $\mathcal{D}$.*

Example 4.3.3. The textual description *"Blast calculates protein sequence similarity"* is pos-tagged as:

```
[NP Blast_NN] [VP calculates_VBZ] [NP protein_NN sequence_NN similarity_NN]
```

The considered text segment is *calculates protein sequence similarity*, since it conforms to the structure (VP, NP+), and its corresponding c-chunk is: {W1107659, E0002979, D9000419, C0002518, D9000518, C1710052, E0001413, C1441506}.

Let us describe the estimation of the proposed translation model. Firstly, an initial concept posterior probability conditioned on the vocabulary concepts is calculated following the method proposed in [35]:

$$p(c_i|c_j) = \frac{p_1(c_i, c_j)}{p_1(c_j)} \tag{4.16}$$

$$p_1(c_i, c_j) \propto \sum_{v \in W} p(c_i|v) \cdot p(c_j|v) \cdot p(v), \tag{4.17}$$

$$p_1(c_j) = \sum_{c_i \in V} p_1(c_j, c_i) \tag{4.18}$$

$p(c|v)_{c \in V, v \in W}$ are entailment probabilities that are estimated as follows:

$$p(c|v) = \frac{tf(c, v)}{|v|} \tag{4.19}$$

where $tf(c, v)$ is the number of times c occurs in window v and $|v|$ is the length of v.

Finally, $p(v)$ is the probability of the window v, which is estimated as the inverse of the cardinal of W.

$$p(v) = |W|^{-1} \tag{4.20}$$

So, the probability $p_1(c_i|c_j)\ \forall c_i, c_j \in V$ can be seen as the probability of translating c_j into c_i in one translation step. Then, $p(c_i, c_j)$ is defined as the smoothed version of $p_1(c_i, c_j)$ obtained by generating random Markov chains between words. It is defined

as follows [71]:

$$p(c_i, c_j) = ((1 - \alpha) \cdot (I - \alpha \cdot P_1)^{-1})_{i,j} \tag{4.21}$$

I is the $n \times n$ identity matrix, P_1 is a $n \times n$ matrix whose element $P_{i,j}$ is defined as $p_1(c_i, c_j)$, and α is a probability value that allows the generation of arbitrary Markov chains between words.

Finally, to complete and rank the set of keywords of facet f, we use the Bayes formula on $p_f(c)_{c \in V}$. That is,

$$p(f|c) \quad \propto \quad \frac{p_f(c)}{p(c)} \tag{4.22}$$

where $p(c)$ is estimated from the linear equation system given by the n variables $p(c_i)_{c_i \in V}$, and $n + 1$ equations:

$$p(c_i) = \sum_{c_j \in V} p(c_i|c_j) \cdot p(c_j) \quad (i \in 1, \cdots, n) \tag{4.23}$$

$$\sum_{c_i \in V} p(c_i) = 1 \tag{4.24}$$

Once K_f and V_f have been defined for each facet f, both sets are used to extract relevant information about the facets from textual descriptions.

Table 4.3 shows the initial keywords and the top-ranked values for the input, output and method facets in the Life Sciences domain. As it can be shown, facets are not mutually exclusive, since a same concept can represent a value of several facets. In a textual description, the facet to which the concept refers can be determined on base to the context in which it appears.

Facet	**Initial keywords**	**Top-ranked values**
Input	Input, take, giving, import, receive	parameter, argument, collection, maps, job, min ID, message, status job, form, ...
Output	Output, search, predict, return, extract, retrieve	output, job, id reference, xref, information, request, sequence, datatype, parameter, format, ...
Method	Method, implement, apply, run, method, perform	method, algorithm, bioinformatics algorithm, signalp, association, technique, centric method construction, application, filter, physical process, ...

Table 4.3: Initial keywords and top-ranked values for the input, output and method facets.

Given a textual description d, the probability that a concept $c \in concepts(\overrightarrow{d})$ represents the probabilistic facet f is estimated from its c-chunks as follows:

$$p(c, f|d) = \sum_{Tc \in c\_chunks(d)} p(c, f|Tc) \cdot p(Tc|\overrightarrow{d}) \tag{4.25}$$

The probability of Tc, $p(Tc|\overrightarrow{d})$, is estimated with the product of its members:

$$p(Tc|\overrightarrow{d}) = \prod_{c_i \in Tc} p(c_i|\overrightarrow{d}) \tag{4.26}$$

where $p(c_i|\overrightarrow{d})$ is the relative frequency of the concept c_i in the semantic vector $\overrightarrow{d}$.

The probability $p(c, f|Tc)$ is calculated as:

$$p(c, f|Tc) = \frac{p(c, f, Tc)}{p(Tc)} \tag{4.27}$$

where

$$p(c, f, Tc) \propto p(f|c, Tc) \cdot p(c|Tc) \cdot p(Tc) \tag{4.28}$$

Therefore,

$$p(c, f|Tc) \propto p(f|c, Tc) \cdot p(c|Tc) \tag{4.29}$$

We could estimate $P(f|c, Tc)$ directly from the c-chunk but due to their small size, we should also take into account the global statistics about the feature. Therefore, the probability $p(f|c, Tc)$ in the chunk represented by Tc is calculated as:

$$p(f|c, Tc) = \alpha \cdot p'(f|c) + (1 - \alpha) \cdot p''(f|c, Tc) \tag{4.30}$$

where $p'(f|c)$ is the probability that the concept c describes the facet f independently of the context in which it appears, and $p''(f|c, Tc)$ is the probability that the concept c describes the facet f taking into account its context in Tc, (i.e., the concepts that appear in Tc).

The probability $p'(f|c)$, which is independent of the context, is defined as:

$$p'(f|c) = \frac{p_f(c) \cdot p(f)}{p(c)} \tag{4.31}$$

The probability $p''(f|c, Tc)$ is estimated as:

$$p''(f|c, Tc) = \frac{p_f(c, Tc) \cdot p(f)}{p(c, Tc)} \tag{4.32}$$

where $p_f(c, Tc)$ is the probability that the facet f is described by Tc and, $p(c, Tc)$ is the total probability of Tc. These probabilities are calculated as follows:

$$p_f(c, Tc) = \prod_{c_j \in Tc} p_f(c_j) \tag{4.33}$$

$$p(c, Tc) = \sum_{f_i \in \mathcal{F}} p_{f_i}(c, Tc) \tag{4.34}$$

Finally, replacing the previous expressions in Formula 4.25, the approximation of the probability $p(c, f|d)$ results in:

$$p(c, f|d) \propto \sum_{Tc \in c\_chunks(d)} p(f|c, Tc) \cdot p(c|Tc) \cdot p(Tc|\overrightarrow{d}) \tag{4.35}$$

Complexity. Regarding the complexity of creating these probabilistic models, the models $p_f(c)$ (Formula 4.14) and $p(c|f)$ (Formula 4.22) are calculated for all the concepts in the reference KRs, and they require $O(N_c)$ storage, considering that the number of facets is constant w.r.t. to N_c. The time complexity of $p_f(c)$ is $O(N_c^2)$, whereas $p(c|f)$ is $O(N_c)$ since it is calculated applying Bayes on $p_f(c)$. Both models rely on a translation model (Formula 4.16) in which the co-occurrences of the concepts have been estimated considering all the metadata of the resources registered in BioCatalogue, myExperiment and SSWAP. This translation model is in $O(N_c^2)$. However, due to the high sparseness of the translation matrix, it requires much less space than N_c^2. Although creating these models has a high computational cost, they are created only once, thus they do not affect the process of identifying facets values given a specific textual description. In order to make their access more efficient, the translation model is stored in a dictionary, and $p_f(c)$ and $p(c|f)$ are stored in inverted files.

With respect to the identification of concepts as facets values given a textual description, it is made by estimating the probability of each concept in the description to be a facet value (Formula 4.35), which consists in local calculations, which are mainly counting, and accessions to global values, like $p_f(c)$, which are stored in inverted files. Therefore, the cost of calculating the probability of a concept to represent a facet value

is constant w.r.t. N and N_c.

4.4 Normalization of Data

The normalization process proposed in this chapter is applied to all the data involved in the discovery process, that is, the user's requirements specification and the resources metadata. Next sections briefly describe how the normalization is applied to these data.

4.4.1 Normalization of User Requirements

The user's requirements specification can be represented in different formats, as explained in Section 3.1, and the information must be extracted in order to make the discovery independent of the characteristics of the technique used to specify them.

Table 4.4 shows the results of the extraction module of each requirements specification technique and the corresponding normalization process. In the simplest formats, the information is just a brief description of the user's needs, but in more complex formats such as SPARQL, i^* or graphs, the information has to be extracted also from the syntaxis of the specification (e.g., SPARQL) and from the underlying structure of the specification (e.g., i^* models, query graphs). In SPARQL, the subject and the object of the triples are entities than can be represented by concepts from KRs, and the predicate, which relates the subject and the object, can describe the functionality or relationships such as a facet-value. In a graph-based specification, the nodes represent entities that can be annotated with concepts, and the edges represent relationships that can describe either the functionality or relationships such as the facet-value relationships. For example, in the i^* model, the text of the *task* elements is the description of the tasks that must be performed by web resources. The relationship between these tasks can implicitly define facets values. For example, the output of a task is the input of the following task with which is related.

Independently of the format in which requirements are specified, the extracted information is normalized. First, the extracted information is semantically annotated with concepts from external knowledge resources. Then, it is characterized using topic-based models determining the relevance of the concepts. Finally, the information is analyzed to automatically identify relevant information about the facets, which is combined with the facets information explicitly described in the requirements specification,

Specification Format	Extracted Information	Normalization Process
Keyword-based search	Set of keywords	N1, N2, N3
Textual description	Textual description	N1, N2, N3
Navigational search	Selected categories	N1, N2
Filtered search	Pairs filter-value, explicit facet values	N1, N2
SPARQL	Entities, property-value, explicit facet values	Ontology alignment
Graphs	Entities, relations, explicit facet values	N1, N2, N3
i^* framework	Tasks, explicit facet values	N1, N2, N3

Table 4.4: Normalization process of different user requirements specification techniques. (N1: Semantic annotation, N2: Characterization, N3: Implicit facets)

e.g., the facet-value pairs described by properties or predicates, or those defined by the pairs filter-value in the filtered search. As a result of the normalization process, the requirements specification is represented by a semantic vector with the concepts describing the whole requirement and a set of semantic vectors representing the user-defined facets. The cost of normalizing the user's requirement specification is constant w.r.t. N_c, due to the use of inverted files in the semantic annotation and knowledge extraction processes.

Currently, our discovery approach supports the following requirements specification formats: keyword-based, textual descriptions and i^* models.

4.4.2 Normalization of Resources Metadata

The resource metadata are normalized to make the discovery independent of their characteristics. Independently of the format in which metadata are stored, they are semantically annotated with concepts from KRs and, then, knowledge extraction techniques are applied to better characterize the resource. The cost of the normalization of a resource metadata is constant w.r.t. N_c.

In our approach, web resource metadata consist of textual descriptions, tags and categories. However, other formats could be supported, like RDF. To normalize meta-

data stored in RDF, the extraction module has to identify the information about the resource, that is, the triples (<*subject-predicate-object*>) relevant for discovery. For each RDF triple, the subject and the object should be aligned to concepts of the reference KRs, and the predicate should define relationships such as facet-value and functionality. All this information would be represented with concepts and, in the case of the facet-value relationships, the value can be automatically assigned to the facet. Notice that as both the RDF specification and the KR are semantically expressed, the normalization process just consists in aligning both vocabularies, and interpreting the predicates of the RDF specification. Table 4.5 shows the process of normalization of each metadata format.

Metadata Format	**Extracted Information**	**Normalization Process**
Textual description	Textual description	N1, N2, N3
Tags	Set of tags	N1, N2, N3
Facets	Pairs facet-value	N1, N2
Categories	Set of categories	N1, N2
RDF	Entities, properties, explicit facet values	Ontology alignment

Table 4.5: Normalization process of different resources metadata formats. (N1: Semantic annotation, N2: Characterization, N3: Implicit facets)

4.5 Conclusions

Data in Life Sciences present a high level of heterogeneity that hinders the matching of information. This problem is even worse in textual descriptions, which present ambiguity and implicitness issues.

In order to reconcile requirements and resources, independently of how the information is represented, we have proposed a normalization process to alleviate the problems of heterogeneity, ambiguity, and implicit information. The normalization process consists of two phases: *(i)* semantic annotation and *(ii)* knowledge extraction.

The semantic annotation of textual descriptions addresses the problem of heterogeneity and ambiguity of data, and improves the reconcilement between resources metadata and user's requirements. The annotation is carried out by an automatic and un-

supervised semantic annotator that is capable of using several KRs to cover as much as possible the different terminologies used in the descriptions, and also to address other problems such the ambiguity.

Then, knowledge extraction techniques are proposed to automatically identify relevant information implicitly described in the textual descriptions. First, we have proposed a topic-based model to estimate the relevance of concepts in the descriptions, determining which concepts best describe the resource. Then, information about some user-defined features is automatically identified using semantics and a probabilistic model.

As a result of the normalization process, the data are enriched with formal knowledge, characterized by means of facets and relevant concepts, and represented in a machine-readable format.

Chapter 5

An IR Model for Web Resource Discovery

This chapter proposes an Information Retrieval (IR) model to discover the web resources that provide the functionality required by the user, and to rank them on base to their relevance to the user's requirement. To address the limitations of current open registries, the presented IR model is based on both probabilistic models and semantics.

This chapter is organized as follows. Section 5.1 reviews some of the most common IR models in the literature. Section 5.2 proposes the retrieval model used to discover the resources that are supposed to fulfill the user's requirements. Finally, in Section 5.3, some conclusions about the proposed IR model are given.

5.1 IR Models

IR can be generally defined as the activity of obtaining documents relevant to an information need from a collection of documents. In IR, the user supplies a query which describes her information needs. The system prompts to her a list of documents ordered by their relevance to the query, that is, how well each result satisfies the user's information needs. In order to retrieve and rank the documents, IR systems define a retrieval model.

The definition of a retrieval model comprises three elements: *(i)* the representation of the documents, *(ii)* the representation of the queries, and *(iii)* a function that measures the relevance of the documents with respect to a query.

Different frameworks have been proposed in the literature to formalize these three elements, leading to different retrieval models.

In this section, some existing models to perform IR tasks are briefly described.

5.1.1 Boolean Model

The boolean model is based on the set theory and the Boolean algebra. Documents are represented as binary weighted vectors, i.e, $\overrightarrow{d} = (w_{1,j}, w_{2,j}, ...)$, where $w_{i,j} \in 0, 1$.

A query is a boolean expression of index terms, e.g., $Q = k_a \wedge (k_b \vee \neg k_c)$. Let g_i return the weight associated with the index k_i in any vector (i.e., $g_i(\overrightarrow{d}_j = w_{i,j})$). Queries are represented as a disjunction of conjunctive vectors (i.e., in disjunctive normal form -DNF). For instance, the query Q represented by $Q_{dnf} = (1, 1, 1) \vee (1, 1, 0) \vee (1, 0, 0)$, where each component is a binary weighted vector associated with the tuple of terms (k_a, k_b, k_c). These binary weighted vectors are called the conjunctive components ($\overrightarrow{q}$) of Q_{dnf}.

The similarity between a document $\overrightarrow{d}_j$ to a query Q is defined as:

$$sim(\overrightarrow{d}_j, Q) = \begin{cases} 1 & \text{if } \exists \overrightarrow{q}_{cc} \text{ in } Q_{dnf} | \forall k_i, g_i(\overrightarrow{d}_j) = g_i(\overrightarrow{q}_{cc}) \\ 0 & \text{otherwise} \end{cases} \quad (5.1)$$

The Boolean model predicts that a document $\overrightarrow{d}_j$ is relevant when $sim(\overrightarrow{d}_j, Q) = 1$. Only documents that strictly satisfy the boolean expression are deemed to be relevant. Otherwise, the document is considered to be non-relevant. This represents the major drawback of the model, since there is no partial matching nor relevance ranking. This approach frequently returns either too few or too many documents in response to a user query. Nowadays, it is well known that (non-binary) index term weighting leads to substantial improvements in retrieval performance.

5.1.2 Vector Model

The vector model [95] represents both documents and queries as vectors in a high dimensional space, i.e. $\overrightarrow{d}_j = (w_{1,j}, w_{2,j}, ..., w_{t,j})$ and $\overrightarrow{Q} = (w_{1,q}, w_{2,q}, ..., w_{t,q})$, where t is the total number of different index terms in the collection. Now, the weights are considered positive and non-binary, i.e., $w_{i,j} \geq 0$. To compute the similarity degree between the query and document vectors, different measures have been proposed in the literature. The most widely used measure is the cosine of the angle formed by these

vectors:

$$sim(\overrightarrow{d}_j, \overrightarrow{Q}) = \frac{\overrightarrow{d}_j \cdot \overrightarrow{Q}}{|\overrightarrow{d}_j| \times |\overrightarrow{Q}|} \tag{5.2}$$

In this formula, $sim(\overrightarrow{d}_j, \overrightarrow{Q})$ varies from 0 to 1. So, instead of predicting whether a document is relevant or not to a query, the vector model returns a list of documents sorted by their degree of similarity to the query. A document might be retrieved even if it matches the query only partially. A threshold value can be established to discard the documents with a degree of similarity under that threshold.

Index terms weights can be estimated in many different ways [96]. The most popular method is the $tf * idf$ weighting. The $tf * idf$ method assigns a high weight to those index terms that occur frequently in the document, but do not appear in many other documents of the collection. The intuition is that frequent terms within a document are good representatives for the document. In contrast, the terms that occur in many documents are not useful for distinguishing relevant from non-relevant documents.

Formally, let N be the total number of documents in the collection and n_i be the number of documents in which the index term k_i appears. Let $freq_{i,j}$ be the frequency of the term k_i in the document d_j, The term frequency factor, tf, is the normalized frequency $f_{i,j}$ of the term k_i in the document d_j:

$$f_{i,j} = \frac{freq_{i,j}}{max_l freq_{l,j}} \tag{5.3}$$

where l represents any index term mentioned in the document d_j. The inverse document frequency factor, idf is given by:

$$idf_i = log\frac{N}{n_i} \tag{5.4}$$

Finally, the $tf * idf$ weighting scheme assigns the following weight to the term k_i in the document d_j:

$$w_{i,j} = f_{i,j} \times idf_i \tag{5.5}$$

The $tf * idf$ weighting approach of the vector model improves retrieval performance of the boolean model. The partial matching and ranking strategy allow the retrieval of documents that approximate the query conditions. The main disadvantage of the vector model is that no formal framework is provided to calculate the index term weights.

5.1.3 Language Modeling

Language modeling considers each document as a language model d_j. The documents are ranked according to $P(Q|d_j)$, that is the probability of obtaining the query Q when randomly sampling from the respective language model.

The calculation of $P(Q|d_j)$ differs from model to model. In the simplest case, each query term $Q = (q_1, q_2, ..., q_m)$ is assumed to be independent of the other query terms, so that the probability can be estimated by:

$$p(Q|d_j) = \prod_{q_i \in Q} p(q_i|d_j) \tag{5.6}$$

After the specification of a document prior $p(d)$, the *a posteriori* probability of a document is used to rank the documents in the collection and it is given by:

$$p(d_j|Q) \propto p(Q|d_j) \cdot P(d_j) \tag{5.7}$$

The probability $p(q_i|d_j)$ can be smoothed to discard non-zero values as explained in Section 4.3.1 or to incorporate a semantic smoothing into the language model. [12] estimates translation models $t(Q|w)$ for mapping a document term w to a query term q_i. Using translation models, the document-to-query model becomes:

$$P(Q|d_j) = \prod_{q_i \in Q} \sum_{w} t(q_i|w) \cdot P(w|d_j) \tag{5.8}$$

[56] proposes a language model method that takes into account information about the context and about user's feedback.

5.1.4 Topic-based Models

In topic-based models [101], explained in Section 4.3.1, relevant documents are retrieved by calculating the similarity between the topic distributions T corresponding to the query Q and to each candidate document d_j using a distributional similarity function. [19] models Information Retrieval as a probabilistic query to the topic model, i.e. the most relevant documents are those that maximize the conditional probability of the query. Given the candidate document d_j, the conditional probability $p(Q|d_j)$ is

calculated by:

$$p(Q|d_j) = \prod_{w_i \in Q} p(w_i|d_j) = \prod_{w_i \in Q} \sum_{t=1}^{T} p(w_i|z=t) \cdot p(z=t|d_j) \tag{5.9}$$

This model emphasizes similarity through topics, with relevant documents having topic distributions that are likely to have generated the set of words associated with the query.

In the literature several topic-based models have been proposed, e.g., probabilistic Latent Semantic Indexing (pLSI) [44], Latent Semantic Analysis (LSA) [57] and Latent Dirichlet Allocation (LDA) [17]. Next section describes the most popular applications of LDA as topic-based model.

5.1.4.1 LDA-based Document Model

LDA has been largely used in the literature as topic-based model for many tasks, e.g., documents categorization [16; 118], tagging [113], document retrieval [107], displaying of search results [41] and faceted search [20; 64; 114] among others. Document modeling and faceted search are the two applications that are of interest to our approach.

With respect to document modeling, LDA can be used as the representation model of the documents content but, as [107] claimed, it is not recommendable to use LDA as the only representation model since it is hardly limited to a predefined number of topics and, therefore, it may not cover all information aspects. [107] proposes different combinations of LDA with other representation models to not depend on the LDA topics. The combination that presents better results for their experiments is:

$$p(w|d) = \lambda(\frac{N_d}{N_d+\mu} \cdot p_{ML}(w|d) + (1 - \frac{N_d}{N_d+\mu}) \cdot p_{ML}(w|\mathcal{G})) + (1-\lambda) \cdot p_{lda}(w|d) \tag{5.10}$$

where $p_{ML}(w|d)$ is the maximum likelihood estimate of word w in document d, and $p_{ML}(w|\mathcal{G})$ is the maximum likelihood estimate of word w in a background collection $\mathcal{G}$. Then, p_{lda} is estimated as:

$$p_{lda}(w|d, \theta^{'}, \phi^{'}) = \sum_{z=1}^{K} p(w|z, \phi^{'}) \cdot p(z|\theta^{'}, d) \tag{5.11}$$

where K is the set of topics, $\theta^{'}$ and $\phi^{'}$ are the posterior estimates of θ, the multinomial

distribution over topics for each document, and ϕ, the multinomial distribution for each topic, respectively.

However, LDA requires to specify the number of latent topics, which is usually hard to know a priori. In addition, some learned topics can be less coherent, less interpretable and less useful than other learned topics, which hampers the quality of the learned topics and, consequently, the trust of users on them. Currently, there are many efforts to measure topics quality, (e.g., [65; 69; 100; 105]), with the aim of improving the topic learning process.

With respect to the faceted search, LDA is used to extract and model the topics described in documents and use them for the faceted search, e.g., [20; 64; 114]. All these approaches consider the topics as facets, while facets in our approach correspond to user-defined relevant features such as input/output parameters which cannot be identified by using LDA.

5.1.5 IR and Life Sciences

In the Life Sciences, IR is usually based on the boolean model. For example, in biomedical literature discovery, PubMed performs a search based on a strict boolean model that does not provide any ranking. This strict search depends heavily on the terminologies used, and moreover it does not consider the relevance of each keyword when retrieving. Lately, several attempts have been done to improve this literature search by means of semantics, e.g., defining ontology query models [112] and expanding queries with new concepts [21; 45] among others.

With respect to web resource registries in Life Sciences, these are also usually based on the boolean model. They represent the resources metadata and the user's requirements as bags of words and perform a boolean search. There are some registries (e.g., Magallanes, myExperiment and BioRegistry) that even allow users to use different boolean operators (AND, OR) in the requirements specification. The limitations of this search have been alleviated by the use of semantics in the web resource registries. The semantic-based registries (e.g., SADI, SSWAP, myExperiment, and BioMoby Cardioshare) store the resources metadata in RDF format, and support semantic requirements specifications, like SPARQL and semantic query graphs, which provide a more specific representation of the requirements. In these registries, the discovery consists in the semantic alignment between the SPARQL query or the semantic query graph and the metadata stored in RDF.

Regarding the final result prompted to the user, no registry provides a ranked list of the retrieved resources with information about their relevance to the user's requirement. However, some registries provide additional information about the resources in order to help the user in the selection. For example, the Bioinformatics Link Directory visualizes information about the strength and the popularity of each resource, as well as "related citations". Other registries provide recommendations apart from the list of discovered resources. For example, in BioCatalogue and myExperiment, once the user has selected a resource, the system visualizes resources that are similar to that selected on base to their categories.

In conclusion, syntactic-based IR in the Life Sciences is based on the boolean model, which is heavily dependent on the characteristics of data, which does not provide a ranked list of the results, and in which all keywords have the same relevance. Next section presents our proposal to address all these issues.

5.2 Web Resource Discovery Model

In this section, we propose an IR model for the discovery of web resources, which is independent of the characteristics of data, i.e., the structure, vocabularies and formats, and which ranks the resources on base to their suitability to the user's requirements. First, we describe the representation of both user's requirements and web resources metadata. Afterwards, we describe how the relevance of a resource to a specific requirement is estimated. Finally, we develop the proposed discovery method.

5.2.1 Data Representation

The representation of the data involved in the discovery is one of the main characteristics that define an IR model. In the proposed model, both the user's requirements specification and the resources metadata are normalized and represented with a set of semantic vectors: one semantic vector representing all the normalized data and one semantic vector per each user-defined facet.

Given a user's requirements specification, the result of its normalization is represented as follows:

Definition 5.2.1. *Let $\overrightarrow{Q}$ be the semantic vector representing the normalization of the user's requirements specification Q. Let define $\overrightarrow{fq_i} = \{c_1 : p_1, ..., c_n : p_n\}$ as the semantic vector associated to the facet f_i in which c represents a concept and p is the probability*

that c represents a value of the facet f_i in the user's requirement specification. Let $FQ = \{\overrightarrow{fq_i}\}_{1 \leq i \leq k}$ be the set of semantic vectors representing the facets values defined by the user in her requirement specification.

The semantic vector $\overrightarrow{Q}$ is generated from the semantic annotation of the user's requirement specification as described in Section 4.2. Then, the semantic vectors $\overrightarrow{fq_i}$ associated to the user-defined facets contain the facets values and their probabilities automatically identified by the facets extraction techniques explained in Section 4.3.2.

On the other hand, web resources metadata are also normalized and represented by a set of semantic vectors as follows:

Definition 5.2.2. *Let $\overrightarrow{r}$ be the semantic vector representing the metadata of the resource r. Let define $\overrightarrow{fr_i} = \{c_1 : p_1, ..., c_n : p_n\}$ as the semantic vector associated to the facet f_i in which c is a concept and p is the probability that c represents a value of the facet f_i in the resource metadata. Let define $FR = \{\overrightarrow{fr_i}\}_{1 \leq i \leq k}$ as the set of semantic vectors representing the facets values described in the web resource metadata.*

Definition 5.2.3. *Let define $\mathcal{WR}$ as the set of all web resources available for discovery. Let $\mathcal{R} = \{(\overrightarrow{r_j}, FR_j)\}_{1 \leq j \leq n}$ be the set that contains the semantic vectors of all web resources in $\mathcal{WR}$.*

These semantic vectors are generated in the same way as the vectors representing the user's requirements specification.

5.2.2 Web Resource Relevance Calculation

The relevance of a resource to a specific user's requirement must not be estimated by the number of words the requirement specification and the web resource metadata have in common, because it would depend on the variability of words, the used vocabularies, the characteristics of the text such as the length, verbosity and so on. The relevance of a resource can be considered as how well the resource fulfills the requirement of the user, considering the functionality of the resource and its features.

The vector model determines the relevance of a resource on base to the similarity of the corresponding semantic vectors. However, although the proposed web resource discovery is based on the vector model, its relevance model is not appropriate since not all concepts have the same relevance when describing the resource and, therefore, they cannot be considered equally when calculating the similarity. For example, in

the keywords-based query *protein sequence alignment*, resources that contain *protein* or *sequence* or *alignment* or a combination of them are discovered. The relevance of a resource is usually estimated on base to the number of matching words, that is, the resources that contain the three words will have the highest relevance score, followed by the resources matching two words. However, the resources that match *protein sequence* should have a lower score than those matching *protein alignment*, since *alignment* represents the information need of the researcher. So, in order to consider the relevance of concepts in the resource characterization when determining the suitability of a resource to the user's requirement, we use the topic-based model described in Section 4.3.1.

Moreover, apart from the importance of the matched concepts, the suitability of a resource also depends on the fulfillment of the facets defined by the user and, therefore, the relevance function must also consider how well the resource fulfills the user-defined facets.

We propose a relevance function that estimates the suitability of a resource by measuring the degree of semantic mapping between the requirements specification and the resource characterization, taking into account the relevance of concepts and the resource features.

Definition 5.2.4. *The function* $relevance(Q, r)$ *determines the relevance of a retrieved resource* r *to the user's requirements* Q*, represented by* $\overrightarrow{Q}$ *and* FQ*. The relevance function is defined by the linear combination described with the formula:*

$$relevance(Q, r) = \alpha \cdot sim(\overrightarrow{Q}, \overrightarrow{r}) + (1 - \alpha) \cdot sim\_facets(FQ, FR) \tag{5.12}$$

where α *determines the weight of the facets fulfillment in the estimation of the resource suitability.*

Definition 5.2.5. *Let* $sim(\overrightarrow{Q}, \overrightarrow{r})$ *be the similarity between the semantic vector of the user's requirement specification,* $\overrightarrow{Q}$*, and the semantic vector of the resource* r*,* $\overrightarrow{r}$*. The similarity is given by the mixture of topic models :*

$$sim(\overrightarrow{Q}, \overrightarrow{r}) = \prod_{c_i \in \overrightarrow{Q}} \sum_{t_k \in \mathcal{T}} p(c_i|t_k) \cdot p(t_k|\overrightarrow{r}) \tag{5.13}$$

Usually, the probability $p(t_k|\overrightarrow{r})$ is unknown since the resource r does not appear in the set of relevant resources descriptions RT_k of the base task t_k. Applying Bayes, $p(t_k|\overrightarrow{r})$ is estimated as:

$$p(t_k|\overrightarrow{r}) = \frac{p(t_k, \overrightarrow{r})}{p(\overrightarrow{r})} \tag{5.14}$$

Assuming that all web resources in $\mathcal{WR}$ have the same chance to be retrieved, then, $p(\overrightarrow{r})$ is an unknown constant for all web resources. Thus, we can rewrite the formula above as:

$$p(t_k|\overrightarrow{r}) \propto p(t_k, \overrightarrow{r}) \tag{5.15}$$

Thus, the joint probability of resources and base tasks can be estimated as:

$$p(t_k, \overrightarrow{r}) \propto \sum_{c_i \in t_k \bigcap \overrightarrow{r}} p(c_i|t_k) \cdot p(c_i|\overrightarrow{r}) \tag{5.16}$$

where $c_i \in t_k \bigcap \overrightarrow{r}$ are the key concepts of the topic t_k that appear in the semantic vector $\overrightarrow{r}$.

The second function of formula 5.12, $sim\_facets(FQ, FR)$, estimates the relevance of a resource r on base to the fulfillment of the user-defined facets.

Definition 5.2.6. *Let consider $sim\_facets(FQ, FR)$ as the similarity between the facets defined in the user's requirement specification, FQ, and the facets of the resource R, FR. This similarity is estimated with the formula:*

$$sim\_facets(FQ, FR) = \sum_{\overrightarrow{fq_i} \in FQ} (\beta_i \cdot \prod_{c_k \in \overrightarrow{fq_i}} p(c_k|\overrightarrow{fr_i})) \tag{5.17}$$

where β_i is the weight of the facet f_i in the relevance estimation and $\sum \beta_i = 1$. The probability $p(c_k|\overrightarrow{fr_i})$ corresponds to the value of the concept c_k in the semantic vector $\overrightarrow{fr_i}$.

This function allows the system to rank the discovered resources on base to the fulfillment of the required functionality and the user-defined facets.

Complexity. The relevance function consists in accessions to values that have been previously calculated and that are stored in inverted files and dictionaries. Therefore,

its cost is $O(n)$ (being n the number of concepts in the user's requirements specification) that can be considered constant w.r.t. N_c.

5.2.3 Web Resource Discovery Method

The proposed discovery approach relies on the semantic mapping of the normalized representation of both user's requirements specification and web resources metadata. The resources are retrieved by the matching of the concepts of the normalized requirements specification. In case there is a concept that does not match any resource, maybe because it is too specific, the matching is repeated with an ancestor of such concept in order to retrieve resources that are described with a lower degree of specificity. In order to make the semantic mapping faster, we use an inverted file that for each concept stores a list of resources that are annotated with that concept. This inverted file requires $O(N_c \cdot N)$ storage. Finally, the matched resources are ranked on base to their relevance to the user's requirement, estimated with the Formula 5.12. Next, Algorithm 2 states the steps performed for the discovery of the most suitable web resources.

Algorithm 2 Web Resource Discovery

Require: Q: User's requirements specification
  results=[]
  relevance={}
  for c in concepts($\overrightarrow{Q}$) **do**
    resources=$\{\overrightarrow{r} \in \mathcal{R} | c \in concepts(\overrightarrow{r})\}$
    while resources = {} **do**
      c'=ancestor(c)
      resources=$\{\overrightarrow{r} \in \mathcal{R} | c' \in concepts(\overrightarrow{r})\}$
      c=c'
    end while
    Append resources to results
  end for
  for r in results **do**
    relevance[r]=$relevance(Q, r)$ (Formula 5.12)
  end for
  Sort relevance
  return relevance

Complexity. With respect to the complexity of the discovery process, considering a query with n concepts, the cost of the web resource discovery is $O(n)$. Then, the

ranking of the r retrieved resources has a cost of $O(n \cdot r)$, since all the probabilities used in the relevance estimation were calculated previously and stored in dictionaries. Therefore, the overall discovery process has a cost proportional to $O(n \cdot r)$, which can be considered constant w.r.t. N_c and N.

5.3 Conclusions

The discovery in web resource open registries in Life Sciences heavily depends on the requirements specification format as well as on the characteristics of the resources metadata. Moreover, these registries do not provide information about the relevance of the retrieved resources.

In this chapter we have proposed an IR model to discover and rank the web resources that are suitable for user's requirements, independently of the characteristics of data (both the requirements specification and resources metadata). This independence is achieved by using normalized data. The normalization of data, more specifically the semantic annotation, alleviates the problem of heterogeneity and ambiguity of data and, moreover, it allows matching related information described with different level of specificity. Moreover, the information extracted by the knowledge extraction techniques is used as facets to better characterize the resources and, therefore, to fulfill the user-defined resource features.

The web resource discovery is based on the semantic mapping between the normalized representation of the user's requirement and the web resources metadata. Then, the discovered resources are ranked on base to their suitability to the user's requirement taking into account their characterization and the fulfillment of the user-defined facets. In the end, the user gets a ranked list of web resources, in which the top-ranked resources are the most appropriate for her requirements.

Chapter 6

Experiments

This section presents a set of experiments carried out to validate the discovery process proposed in this thesis. Before presenting the results of these experiments, in Section 6.1, we describe the resources and datasets used. In the rest of the chapter, we describe the experiments performed to evaluate each phase of the process and their results. First, in Section 6.2, we evaluate the normalization process, checking first the semantic annotation and, then, the knowledge extraction techniques. After, in Section 6.3, we evaluate the discovery and ranking of resources with a pool of queries that validate the whole discovery process. In Section 6.4, we compare the results with other retrieval models such as LDA and keyword-based retrieval and, in Section 6.5, we compare our approach with the discovery functionality provided by one of the most popular open registries in Life Sciences, BioCatalogue. Finally, in Section 6.6 some conclusions about the experiments are given.

6.1 Experiments Setup

In this section, we describe the two main sets of resources that have been used in the experiments presented in this chapter: the knowledge resources and the web resources metadata.

With respect to the knowledge resources, we have used several KRs in order to cover as best as possible the different terminologies that appear in the web resources metadata. In contrast to other semantic annotation applications, e.g., the annotation of research articles, in which the target data are related to a specific domain, web resources metadata contain terms from different terminologies (mainly Biology, Bioinformatics,

and Computer Science) which are not well covered by a single KR. Moreover, we have realized that even using several of the existing KRs, there are terms that are relevant in our corpus that are still not covered. To cover these terms, we have created an ad hoc lexicon that includes named entities not defined in current KRs, e.g., popular data formats (e.g., PDF, SOAP, REST), the name of resources usually mentioned in resources metadata (e.g., MUSCLE, Clustal), the name of algorithms (e.g., Smith & Waterman, Huang and Millers), and widely used abbreviations (e.g., mol, seq) among others.

To cover the different terminologies, in these experiments we have used five different KRs (described in Section 4.1.1): UMLS (2,268,460 concepts), EDAM (1699 concepts), myGrid (369 concepts in the domain ontology and 66 concepts in the service ontology), a fragment of Wikipedia related to Bioinformatics (566 concepts), and our named entities lexicon (5174 concepts). All these KRs are stored as inverted files to optimize queries during the normalization process.

With regard to the web resources metadata, we have downloaded (through the registries APIs) and stored all the available metadata of the resources registered in three popular registries in the Life Sciences domain (all of them described in Section 2.2): BioCatalogue (more than 2200 web resources), myExperiment (more than 2000 workflows), and SSWAP (more than 2700 web resources). These registries have been selected to carry out the experiments presented in this thesis, but any other web resource registry could be considered to get the available metadata of the registered web resources.

6.2 Normalization Evaluation

The normalization of data is crucial in the discovery of the most suitable web resources, since the proposed discovery and ranking processes are based on the normalization of the involved data. There are two key aspects to be considered in the normalization of data: *(i)* the normalized data must represent as accurate as possible the information described, and *(ii)* the normalized data must describe as most information as possible trying to reduce the lose of information, including that implicitly described.

In this section, we evaluate the normalization process to ensure the quality of the normalized data, which is the input data of the discovery process. The evaluation is performed in two phases. First, we analyze the semantic annotation process to

determine the quality of the semantic annotations and, then, we evaluate the knowledge extraction techniques.

6.2.1 Semantic Annotation Evaluation

The evaluation of the semantic annotation process is carried out by analyzing the annotated resources metadata obtained from BioCatalogue, SSWAP, and myExperiment. The web resources metadata have been annotated with a total of 187,996 semantic annotations (43.6% in BioCatalogue, 39.7 % in SSWAP, and 16.7% in myExperiment). In these annotations, 13,808 different entities are semantically annotated with 14,355 different concepts (15,332 before applying the simplification techniques). Figure 6.1 shows the number of concepts in the original annotation and the number of concepts after applying the simplification techniques. As it can be noticed, the reduction in the number of concepts is related to the specificity of the KR, that is, it is lower as the KR more specific is.

With respect to the distribution of concepts in the registries, Figure 6.2 shows the number of concepts of each KR in the semantic annotation of the metadata of the resources in each registry. The most used KR is UMLS, with 66% of concepts in BioCatalogue, 58.9% of concepts in SSWAP, and 61% in myExperiment. EDAM is more used in the annotations of SSWAP resources metadata than in other registries, since the vocabulary used in SSWAP is more related to Bioinformatics (the domain covered by EDAM ontology). Obviously, the least used KR is the fragment of Wikipedia according to its size.

Regarding the types of annotations on base to the number of words that compose the annotated entities, Figure 6.3 shows the proportion of the different semantic annotations in each registry. Considering the 14,355 different concepts that appear in the normalized metadata of the three registries, the 77.17% of these concepts match single word entities, 19.71% match two words entities, 2.73% match three words entities, 0.37% match four words, and 0.02% match five. Unfortunately, the annotations of single word entities sometimes introduce ambiguity because a single word can have different senses and, depending on the context in which it appears, it has a different sense. The semantic annotator does not differentiate between senses, and selects all the concepts matching the term. In contrast, when the concepts match more than one word, the ambiguity is reduced since an entity described by a combination of words hardly ever has different senses. In order to demonstrate this fact, we have manu-

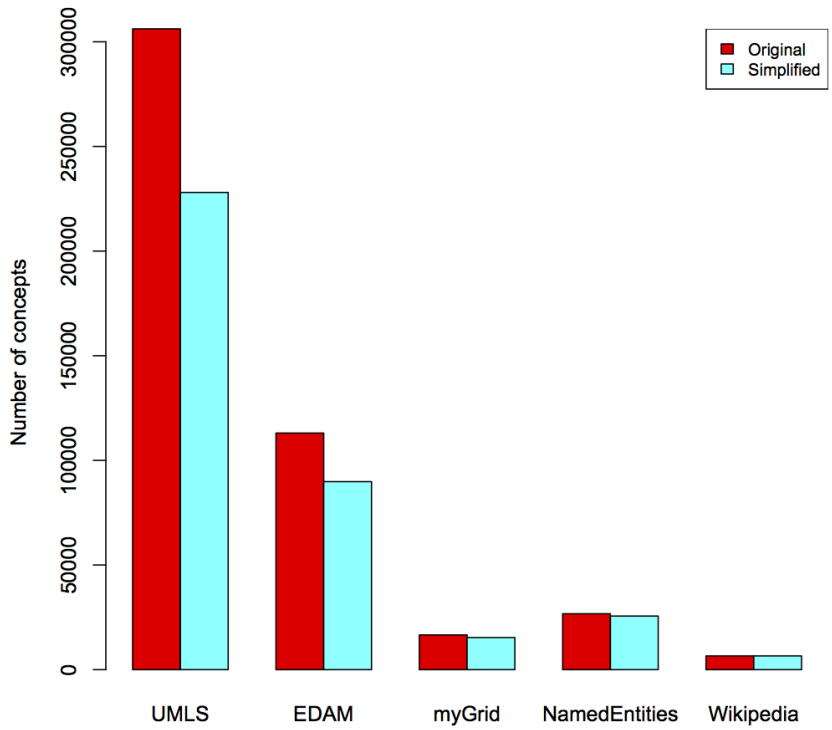

Figure 6.1: Number of concepts of the different KRs in the original semantic annotations and in the simplified annotations.

ally analyzed a sample of two-words entities annotations and a sample of three-words entities annotations to determine how well these multi-word annotations describe the correct sense of the entity. As a result, the 94% of the two-words entities annotations and the 96% of the three-words entities annotations are correct. Therefore, we can say that the ambiguity is mainly present in the annotations of one-word entities, which might introduce noise in the discovery process.

In order to analyze the impact ambiguous annotations have on the discovery process, we have carried out a set of experiments to determine if the matched concepts, those remaining after the simplification, represent the correct sense of the term in the specific context in which it appears and, therefore, determine the precision of the semantic annotations.

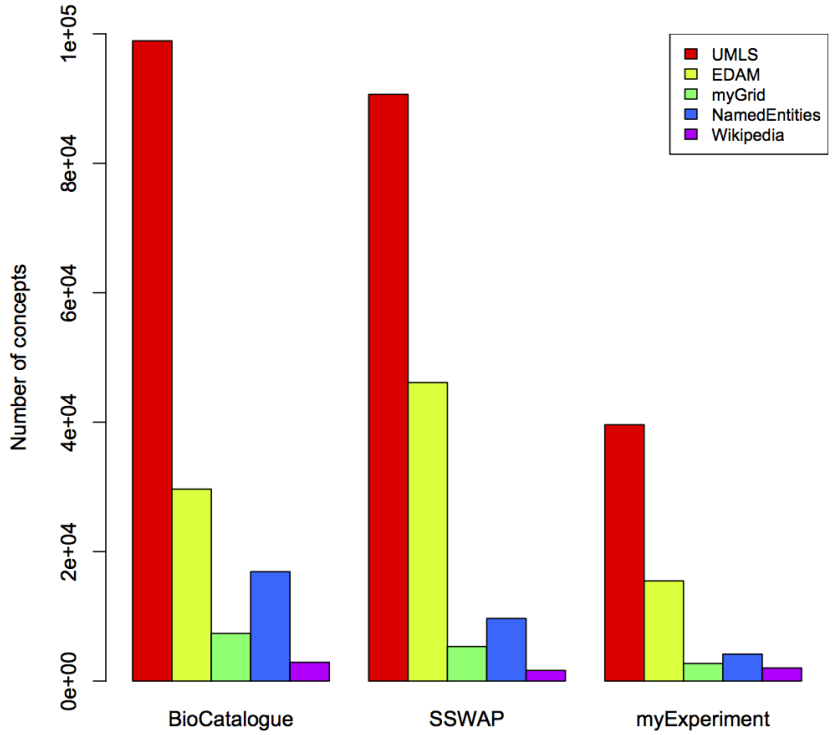

Figure 6.2: Distribution of concepts in the annotations of the metadata registered in BioCatalogue, SSWAP, and myExperiment.

In order to perform this evaluation, we have considered the semantic annotations that match single-word entities and that have concepts of different semantic types, that is, concepts assumed to be describing different senses. Then, we have manually created with them a gold standard (GS) determining the validity of each annotation, taking into account the context in which it appears. To build this GS, we have considered only UMLS concepts since they have well-defined semantic types. For each one of the considered ambiguous semantic annotations, we have stored an entry for each concept in the annotation that consists of: the name of the resource in which the terms appears, the ambiguous term, and the concept. For each entry, we have assigned to it a "1" if the concept describes the correct sense of the term in the resource, or "0" if not. We have manually evaluated 3716 semantic annotations annotating 717 different terms

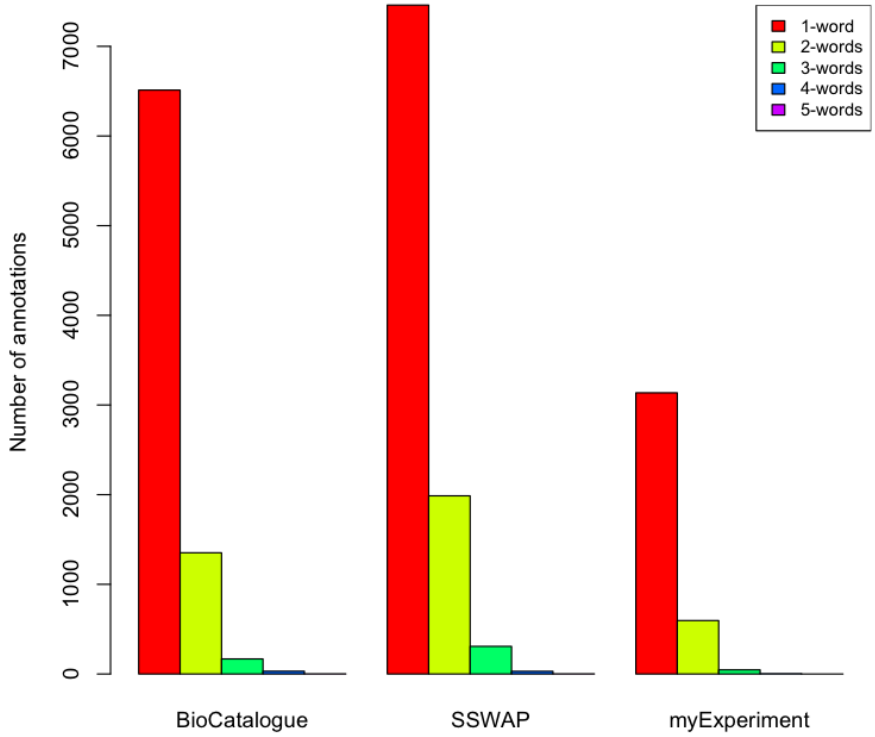

Figure 6.3: Types of annotations depending on the number of matched words.

with a total of 1033 different concepts. There are cases in which the correct sense is not described by any of the matched concepts, that means that there is a *hidden sense* that may be described by a concept from another KR. In our GS, there are 258 terms for which any of its UMLS matched concepts describe the correct sense.

Then, to evaluate the ambiguity degree in the semantic annotations, we have selected the ambiguous semantic annotations and we have evaluated them by calculating two measures: error and precision. We define *error* as the probability that the sense of a term in a specific context is not described by any of its matched concepts. We consider *precision* as the probability that the semantic annotation describes the correct sense of a term in a specific context. To calculate these measures, we assume that the concepts of KRs different from UMLS describe the correct sense of a term, since those KRs are more specific than UMLS.

Apart from calculating the error and precision measures of the ambiguous annotations using all KRs, we have also analyzed the impact of each KR in the quality of

KRs	Error	Precision
All KRs	0.19	0.56
Without named entities lexicon	0.22	0.54
Without Wikipedia	0.2	0.55
Without named entities lexicon and Wikipedia	0.23	0.53
Without myGrid	0.2	0.56
Without EDAM	0.27	0.43
Without EDAM and myGrid	0.28	0.43
Only UMLS	0.31	0.42

Table 6.1: Error and precision measures of the semantic annotations using different combinations of KRs in the semantic annotation process.

the annotations by calculating the error and precision using different combinations of KRs in the semantic annotation process. Table 6.1 shows the error and precision of the semantic annotations using different combinations of KRs.

The overall error is around 0.19, which means that the 19% of the ambiguous annotations do not represent the correct sense of the term in the context in which it appears. On the other hand, 56% of the matched concepts in ambiguous annotations represent the correct sense of the term. As it can be noticed in Table 6.1, when using only UMLS as KR, the 31% of the ambiguous annotations do not represent the correct sense of the term. Using EDAM improves the precision, since the semantic annotations without EDAM present a higher error and a lower precision. It is also worth noting that our lexicon of named entities has also an impact on the annotations, higher than other KRs such as Wikipedia or myGrid.

These results show that using several KRs with different level of specificity alleviates the problem of ambiguity. Moreover, the use of several KRs also addresses the lack of coverage of the different terminologies used in the resources metadata, as we have demonstrated, for example, with the lexicon of named entities.

6.2.2 Knowledge Extraction Evaluation

This section presents the evaluation of the proposed knowledge extraction techniques. First, we describe the topic-based model built for the discovery of resources in the Life Sciences domain. Then, we present the validation of the methods used to extract relevant information implicitly described in textual descriptions.

	Topic
T_1	Search proteins with a functional domain
T_2	Localize protein expression
T_3	Search similar sequences
T_4	Identify and characterize genes linked to a phenotype
T_5	Analyze transgenic model organism
T_6	Find genes with functional relationships
T_7	Find common motifs in genes
T_8	Predict structure
T_9	Identify putative function of gene
T_{10}	Gene prediction
T_{11}	Analyze phylogeny
T_{12}	Align sequences
T_{13}	Protein identification and characterization

Table 6.2: Bioinformatics base tasks defined as topics

6.2.2.1 Resource Characterization Evaluation

The evaluation of the resource characterization consists in validating the topic-based model created to determine automatically the relevance of each concept in the resource characterization.

In Bioinformatics, there are some tasks that are very common, and most taxonomies describing categories of resources are based on these tasks, e.g., BioCatalogue categories or OBRC categories. Regarding these reference tasks and those defined by [103] as relevant bioinformatics tasks, we have defined 13 bioinformatics base tasks as topics, shown in Table 6.2, to build the topic-based model. However, the model can be modified and extended with new tasks.

To build the topic-based model, for each base task t_k we have specified a set of key concepts to retrieve web resources that are relevant to t_k. These concepts can be automatically gathered either from existing documents, such as Wikipedia pages related to each topic, or from well-defined taxonomies of categories. For example, for the topic "search similar sequences", some examples of key concepts are: *KR0000204* and *W363695* (Blast), *E0001413* (Sequence similarity), and *C0162774* (homologous sequences). For each one of the key concepts of each topic, we have automatically retrieved the resources that contain such key concept, and we have selected the top-10 resources ranked by using the cosine measure over their $tf \times idf$ semantic vectors. Once

the sets of relevant resources, called RT_k, have been created, the topic-based model has been built for all concepts in the KRs as explained in Section 4.3.1.1. Figure 6.4 shows the cardinality of each RT_k and the source of the selected resources. As Figure 6.4 shows, there are some topics that are not well-represented in these registries, e.g., T_2, T_4, T_{10} and T_{13}, and, in consequence, their RT_k is very small. We have not evaluated these topics since we consider that their models are not representative enough, due to the low representation. We can also notice that there are some topics that are more frequent in some registries than in others. For example, T_1, T_6 and T_8 are more frequent in SSWAP, while T_3, T_{11} and T_{12} are more frequent in BioCatalogue.

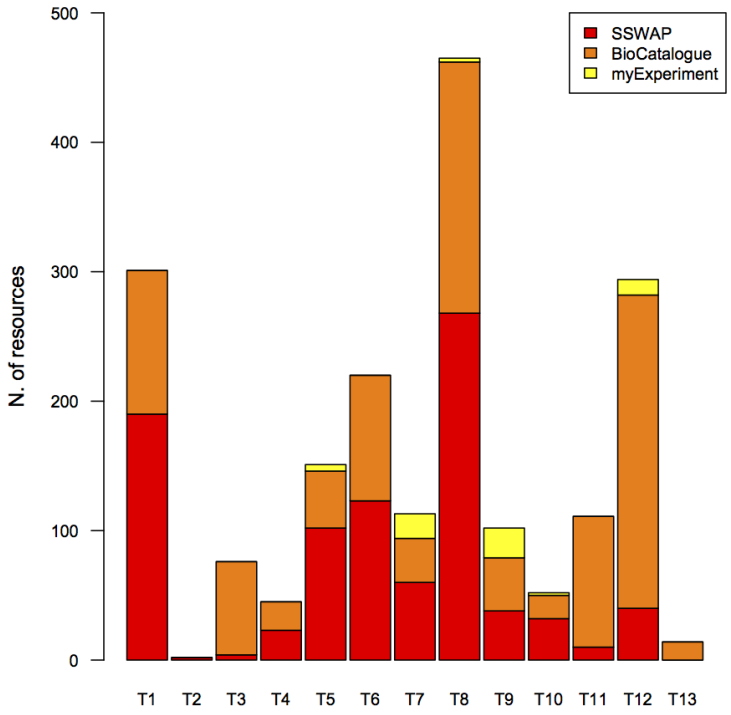

Figure 6.4: Cardinality of the RT$_k$ of the topics.

In Table 6.3, we show the most frequent BioCatalogue categories of the resources in each RT_k, which reflect the correspondency between categories and topics.

The results of the evaluation of the topic-based model are shown in the evaluation of the discovery and ranking process in Section 6.3.

6.2.2.2 Facets Extraction Evaluation

In this section, we show the results of the evaluation of the techniques proposed for the extraction of relevant information about the resources facets that are implicitly described in textual descriptions. First, we present the results of the evaluation of the probabilistic techniques and, then, the results of the semantic-based facets.

Probabilistic facets. In these experiments, we have considered as probabilistic facets the input, the output and the method facets. To carry out this evaluation, we have set up a GS data set with the explicit information about the facets of the resources registered in BioCatalogue. We have selected BioCatalogue as reference source to build the GS since it allows users to assign tags to resources in order to describe features such as the input and the output data types. Currently, 59 resources in BioCatalogue have at least one tag describing its input or its output (48 resources have at least one tag describing its input, and 48 resources have at least one tag describing its output). There are 162 tags (47 different annotated with 88 concepts) describing the input, and 95 tags (46 different annotated with 100 concepts) describing the output of the resources. The most frequent tags for the input are: *fasta format, protein sequence* and *DDBJ record*, and the most frequent tags for the output are: *DDBJ record, gene prediction report* and *BIND record*.

To build the GS, we have differentiated between those facets that have explicit tags describing them, i.e., input and output, and those that are not explicitly mentioned, i.e., method. For input/output facets, we have automatically selected the tags assigned to the input/output descriptions. For the method, we have manually classified the tags. Table 6.4 shows the number of different concepts annotating the tags and the number of involved resources for each facet. It is worth noting that few resources are tagged with input/output descriptions in BioCatalogue, which confirms the lack of processable metadata in this kind of registries. This table also shows the results of our facet extraction technique: the number of initial keywords specified to build the probabilistic model, the number of concepts identified automatically as facets values, and the number of resources that have been tagged with concepts identified as facets values (number of BioCatalogue resources and total number of resources independently

	Topic	**Top BioCatalogue categories**
T_1	Search proteins with a functional domain	Domains
T_2	Localize protein expression	N/A
T_3	Search similar sequences	Protein sequence similarity Nucleotide sequence similarity
T_4	Identify and characterize genes linked to a phenotype	N/A
T_5	Analyze transgenic model organism	Microarrays, Biostatistics Data retrieval
T_6	Find genes with functional relationships	Pathways, protein interaction
T_7	Find common motifs in genes	Function prediction, motifs
T_8	Predict structure	Protein secondary structure Protein tertiary structure Protein structure prediction
T_9	Identify putative function of gene	Functional genomics Function prediction Domains
T_{10}	Gene prediction	Genomics Sequence analysis Gene Prediction
T_{11}	Analyze phylogeny	Phylogeny
T_{12}	Align sequences	Protein sequence alignment Nucleotide multiple alignment Protein multiple alignment Nucleotide sequence alignment...
T_{13}	Protein identification and characterization	Chemoinformatics

Table 6.3: The most frequent BioCatalogue categories in the RT_k of each topic.

	GS		Target Resources			
Facets	Concepts	BioCatalogue Resources	Initial keywords	Facet concepts	BioCatalogue Resources	Resources
Input	88	48	9	1696	1404	4875
Output	100	48	19	2151	1612	5169
Method	295	1316	16	3377	2102	5814

Table 6.4: For each facet, *(i)* characteristics of its GS (number of different concepts annotating the tags and number of BioCatalogue resources tagged), and *(ii)* the results of our facet extraction method: initial keywords for the probabilistic model and the number of concepts and tagged resources identified by our method (BioCatalogue resources and resources from the three registries).

of their source). As it can be observed, the number of identified concepts is higher than those in the GS, and the number of tagged resources is also considerably higher than those identified in the GS, since our approach identifies facet values in all the available metadata, that is, not only in the specific fields for the facet, but also in textual descriptions.

To evaluate the quality of the extracted facets values, we have calculated the precision, recall and F-measure of the results as it is next explained.

For a given facet f_i, we denote with $tags(f_i)$ the BioCatalogue tags in the GS assigned to f_i, and with $concepts(f_i)$ the automatically extracted concepts for facet f_i. Each tag $t \in tags(f_i)$ has associated the set of resources annotated with it for the facet f_i, which is denoted with $services_{f_i}(t)$.

Similarly, each concept $c \in concepts(f_i)$ has associated the set of resources having c as value of the facet f_i, denoted as above.

We calculate precision, recall and F-measure for each pair $(t, c), t \in tags(f_i)$ and $c \in concepts(f_i)$, as follows:

$$P_{f_i} = \frac{services_{f_i}(t) \cap services_{f_i}(c)}{services_{f_i}(c)} \tag{6.1}$$

$$R_{f_i} = \frac{services_{f_i}(t) \cap services_{f_i}(c)}{services_{f_i}(t)} \tag{6.2}$$

$$F_{f_i} = 2 \cdot \frac{P_{f_i}(t,c) \cdot R_{f_i}(t,c)}{P_{f_i}(t,c) + R_{f_i}(t,c)} \tag{6.3}$$

Facet	Precision	Recall	F-measure
Input	0.74	0.91	0.77
Output	0.65	0.96	0.72
Method	0.93	0.51	0.62

Table 6.5: Precision, recall and F-measure of the probabilistic facets considering the GS.

The global precision and recall is calculated as a macro-average over the best (t, c) mappings, which is defined as:

$$P_{f_i} = \sum_{t \in tags(f_i)} P(t, argmax_{c \in concepts f_i}(F_{f_i}(t, c))) \cdot \frac{1}{|tags(f_i)|} \tag{6.4}$$

$$R_{f_i} = \sum_{t \in tags(f_i)} R(t, argmax_{c \in concepts f_i}(F_{f_i}(t, c))) \cdot \frac{1}{|tags(f_i)|} \tag{6.5}$$

Table 6.5 presents the values of these measures for the facets extracted with the probabilistic technique, i.e., input, output and method. The results reveal that this technique obtains, in general, good effectiveness for the facets input, output and method. For input and output facets, whose GS is based on well-defined metadata, the high recall states that almost all facets values explicitly specified as facets in the resources metadata are also identified by our method. On the other hand, the precision of the method facet shows that our method identifies correctly the values of such facet.

Semantic facets. In these experiments, we have considered as facets the species and the diseases the information provided by the resource is related to. For each facet, we have selected all concepts whose semantic type is associated to the facet, and we have validated manually each concept. We have considered only UMLS concepts since, as mentioned before, UMLS is the only KR, from those used in these experiments, with well-defined semantic types. Table 6.6 shows the number of concepts considered as facets values, the number of resources that have been tagged with these facets values, and their precision. The precision of both facets is good, although the precision of the facet disease reflects that further studies about the semantic types assigned to this facet must be done in the semantic annotator, since many concepts are not correctly assigned. In this experiment we have not calculated the recall since it would require to analyze which concepts are related to these facets, but whose semantic types do

Facet	Concepts	Resources	Precision
Disease	137	178	0.74
Species	292	809	0.82

Table 6.6: Number of concepts, number of tagged resources, and precision of the semantic facets.

not describe them correctly. In this case, instead of evaluating our method, we would evaluate the quality of the concepts categorization in UMLS, which is out of the scope of this thesis.

6.3 Discovery and Ranking Evaluation

The evaluation of the discovery and ranking process is, in some way, the evaluation of the whole proposed approach, since both discovery and ranking depend on the previous steps. Therefore, we can consider this evaluation as the evaluation of the whole proposed discovery system.

The experiments carried out to perform this evaluation consist in the execution of a set of heterogeneous queries (i.e., task description examples) that capture different ways to describe bioinformatics tasks, thus reflecting the variability in the users' information needs. The query pool, available on the web site[1], was created by selecting more than 250 short descriptions related to the defined topics and extracted from other Life Sciences resource catalogues such as OBRC[2] (Online Bioinformatics Resource Collection) and ExPaSy[3] (SIB Bioinformatics Resource Portal). Both catalogues define a taxonomy of categories that can be related to the topics defined in our topic-based model. So, we have selected as queries short descriptions of resources registered on these catalogues and annotated with the related categories. From this pool of queries, we consider only for evaluation those corresponding to the topics that can be unambiguously described and whose RT_k are representative enough in order to perform an accurate evaluation. Table 6.7 shows the number of selected queries associated to each topic.

To evaluate the results of these queries, we have built a GS, since it is not feasible to

[1] http://krono.act.uji.es/KAIS/pool_queries.xml
[2] http://www.hsls.pitt.edu/obrc/
[3] http://expasy.org

determine the whole set of relevant results for each query. The GS[1] has been built for seven topics that can be unambiguously described, and we have revised it manually in order to ensure the quality of the final set. Figure 6.5 shows the proportion of resources from the different registries that form the GS of each evaluated topic. As shown in Figure 6.5, the GS is mostly composed of SSWAP and BioCatalogue resources, and their proportion is related to the representation of the topic on each registry.

With this GS, we have evaluated the results obtained for each one of the queries from the query pool with the traditional precision, recall and F-measure, shown in Table 6.7. The results show that the top-ranked resources are, in most cases, appropriate for the user's requirement and, moreover, the high recall indicates that usually most of the relevant resources are provided to the user.

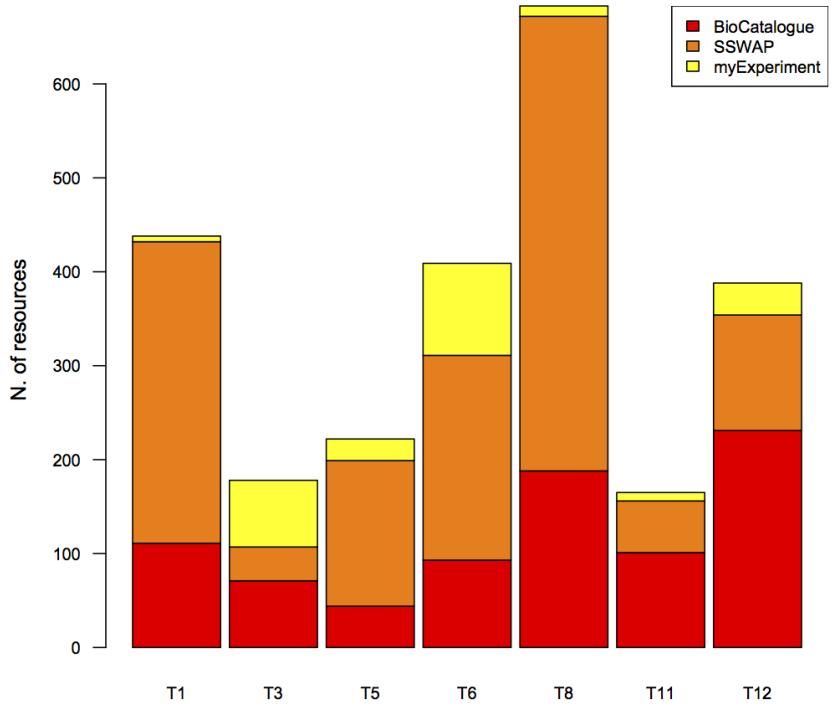

Figure 6.5: Composition of the GS to evaluate the discovery process.

[1]http://krono.act.uji.es/KAIS/gold_standard.xml

	Topic	N. of queries	P@5	P@10	P@20	P	R	F
T_1	Search proteins with a functional domain	14	0.94	0.96	0.94	0.51	0.81	0.63
T_3	Search similar sequences	16	0.81	0.79	0.76	0.19	0.47	0.27
T_5	Analyze transgenic model organism	31	0.92	0.92	0.91	0.6	0.85	0.7
T_6	Find genes with functional relationships	42	0.88	0.84	0.82	0.42	0.43	0.43
T_8	Predict structure	30	0.84	0.84	0.81	0.56	0.41	0.47
T_{11}	Analyze phylogeny	14	0.8	0.81	0.83	0.46	0.89	0.61
T_{12}	Align sequences	24	0.92	0.92	0.88	0.43	0.62	0.5
	Average	24.42	0.87	0.87	0.85	0.45	0.55	0.52

Table 6.7: Precision (P), recall (R), and F-measure (F), including the precision for the top-5, top-10, and top-20 results.

Table 6.8 shows the top-10 resources for the query "Calculate maximum likelihood phylogenies given nucleotide sequences", and it can be shown that most of them are not categorized in their registries. The five top-ranked resources perform phylogeny using maximum-likelihood.

6.4 Comparison with other Retrieval Models

Two of the main characteristics of our discovery process are the normalization of data and the semantic-based mapping. In this section, we compare our discovery process with other techniques used broadly in the literature for similar purposes. Section 6.4.1 presents a comparison between the results obtained in the discovery when using our topic-based model and when using LDA to characterize the resources metadata. In Section 6.4.2, we demonstrate that using semantics in the discovery process improves considerably the results by comparing the semantic-based results with those using a keyword-based discovery.

6.4.1 LDA to Characterize Data

As we have said in Section 5.1.4, LDA can be used as a model to represent the information described in a text. Here, we demonstrate that LDA does not get good results in the discovery of the most suitable resources in the Life Sciences, since it does not

Resource	Registry	Categories
runPhylipDnaml	BioCatalogue	N/A
INB:inb.bsc.es:runPhylipDnamlk	BioCatalogue	N/A
INB:inb.bsc.es:runPhylipDnamlk	BioCatalogue	N/A
TREE-PUZZLE	SSWAP	Phylogeny reconstruction, DNA, protein
runPhylipDnamlk	BioCatalogue	N/A
rociImplementationService	BioCatalogue	Phylogeny
INB:inb.bsc.es:runPhylipProtpars	BioCatalogue	N/A
INB:inb.bsc.es:runPhylipDnapars	BioCatalogue	N/A
fconsense	BioCatalogue	N/A
ftreedistpair	BioCatalogue	N/A

Table 6.8: Top-10 results for the query "Calculate maximum likelihood phylogenies given nucleotide sequences".

model correctly the topics that appear in the resources metadata.

In this experiment, we have used LDA instead of our topic-based model to characterize the resources metadata, and we have compared its results with the results obtained using our topic-based model. We have carried out the experiment setting the number of LDA topics to 13 (the number of topics in our topic-based model) and to 7 topics, with the aim of analyzing the impact of low quality topics (more frequent when the number of topics is higher). In both experiments, the number of iterations has been set to 1000.

Table 6.9 shows the top-10 terms of the topics identified by LDA with 13 topics. As it can be observed, only two topics describe accurately a Bioinformatics task, e.g., topic 1 describes the tasks of phylogeny and sequence alignment (our T_{11} and T_{12}), and topic 2 corresponds to sequence similarity tasks (our T_3). Other topics combine terms from different tasks, e.g., topic 6 mixes terms of microarrays (our T_5) with protein domains (our T_1) and interactions (our T_6), and there are others that are not coherent, e.g., topic 11. In contrast, our topic-based model does not suffer from this problem since the topics are related to well-known bioinformatics tasks, and they are built on base to an initial set of known key concepts relevant for each topic, which are usually related to resources categories.

To compare LDA with our topic-based model, we have executed the pool of queries considering both LDA models, that with 13 topics and that with 7 topics. We have calculated the precision, recall and F-measure for both models. Figure 6.6 compares

Topic 1	**Topic 2**	**Topic 3**	**Topic 4**
sequence alignment	protein sequence	p53	PubMed
tree	blast	id gene	extraction
clustalw	sequence similarity	knowledge repository	data
sequence analysis	nucleotide sequence	gene mutation	phenotype
phylogenetic analysis	sequence analysis	codon	recognition
FASTA	DDBJ	exon	user
phylogenetic	FASTA	ids protein	person
protein sequence	EBI	line	runs
phylogenetic tree	protein sequence analysis	distribution	corpus
t-Coffee	Fasta format	SPS	genome
Topic 5	**Topic 6**	**Topic 7**	**Topic 8**
SOAP	retrieval	sequence analysis	protein domain
Biomart	data	genome	EBI
data	KEGG	sequence analysis	gene ontology
human	interactions	vectors	poll
id gene	ligand	contents	protein
OMIM	protein domain	genome	genomics
SBML	ids gene	output	protein
Ensembl	Ensembl	similarity	bioinformatics
KEGG	enzymes	excel	PDB
Homo sapiens	ENZYME	operation	protein structure
microarray database	microarray database	data	ORF
Topic 9	**Topic 10**	**Topic 11**	**Topic 12**
remove	data	operator	gene express
aida	biogrid	parameters	molecule
genomics	microarray database	user	SMILE
character	literature repository	filtered	user
MEDLINE	experiment	apply	API
remove	max	country	trident
genome	weather	city	sample
tags	sources	learning	atoms
filter	physics-based	applied	split
concepts	repository	values	PUG
Topic 13			
Ensembl			
R processor			
DDBJ			
Swissprot			
molecule			
sequence cluster			
cluster analysis			
disk			
id protein			
sequence			

Table 6.9: Top-10 terms of the LDA topics (k=13)

the results obtained by the two LDA models with the results obtained by our topic-based model. Notice that our approach obtains better results than LDA, with a higher precision in the top-k resources and a higher recall in almost all topics. With respect to the two models of LDA, there is not much difference between their results.

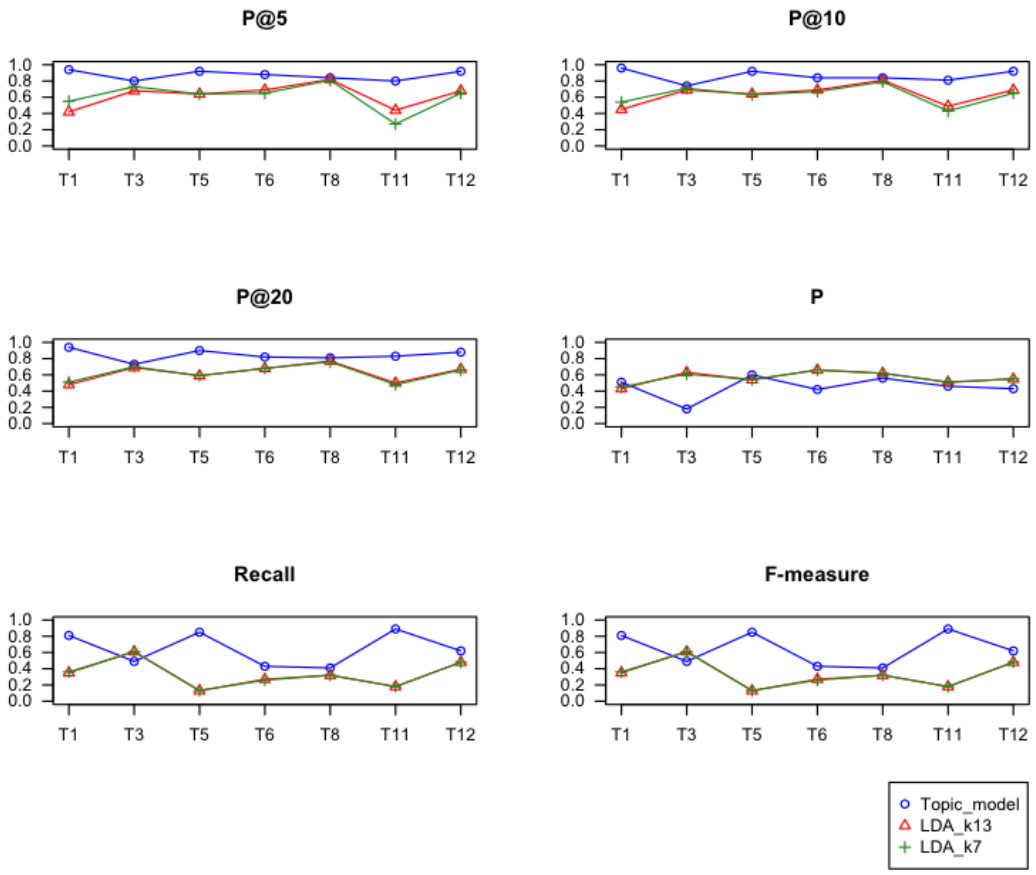

Figure 6.6: Precision of top-5, top-10, top-20, and the overall precision, recall, and F-measure of the results of our topic-based model, LDA with 13 topics, and LDA with 7 topics.

6.4.2 Keyword-based Discovery

Most current open registries base the discovery on the string matching of query keywords on the resources metadata, which hardly consider lexical variants nor even synonyms and hypernyms. To address these limitations, we base the discovery on the

semantic mapping of concepts.

Here, we demonstrate that the semantic-based discovery presents better results than the discovery based only on keywords. We defined two experiments: *(i)* discovery based only on keywords, and *(ii)* discovery based on keywords using a topic-based model.

In the first experiment, the discovery consists in the string matching between the words in the user's requirements specification and the words in the resources metadata, without any type of normalization. We have executed the queries from the query pool, and the precision of the discovered resources is in average 32% and the recall is 38%. Therefore, this kind of discovery, the one used by most current open registries, presents a low precision and a low recall, which means that most of the resources provided to the user are non-relevant for her requirement, and not all the relevant resources are discovered. The main reason is that this type of matching requires that the user's requirements specification is expressed with the same words as the resources metadata, and this is almost always unfeasible since users do not know the vocabulary used to describe the resources and, moreover, the resources metadata present a high level of heterogeneity. Therefore, the performance of this type of discovery is usually very poor.

In the second experiment, we have built a topic-based model based on words instead of concepts with which the data are characterized. In this experiment, the discovery is based on this model, and it retrieves relevant resources performing a keyword mapping instead of a concept mapping. To validate this keyword-based approach, we have executed the queries from the query pool using the keyword-based discovery, and we have calculated the precision, recall and F-measure of the results. Figure 6.7 shows the comparison of the precision, recall and F-measure of the results obtained by this lexical discovery and the results obtained by our approach. As shown in the graphs, our semantic-based approach obtains better precision in the top-k results and a higher recall.

6.5 Comparison with other Web Resource Registries

Finally, in order to compare our approach with the search engines of current web resource registries in Life Sciences, we compare it with the BioCatalogue search engine. We have selected BioCatalogue because nowadays it is one of the most popular open registries in Life Sciences, and because BioCatalogue provides an API that allows users to query it programmatically. BioCatalogue provides two types of search: *(i)* keyword-

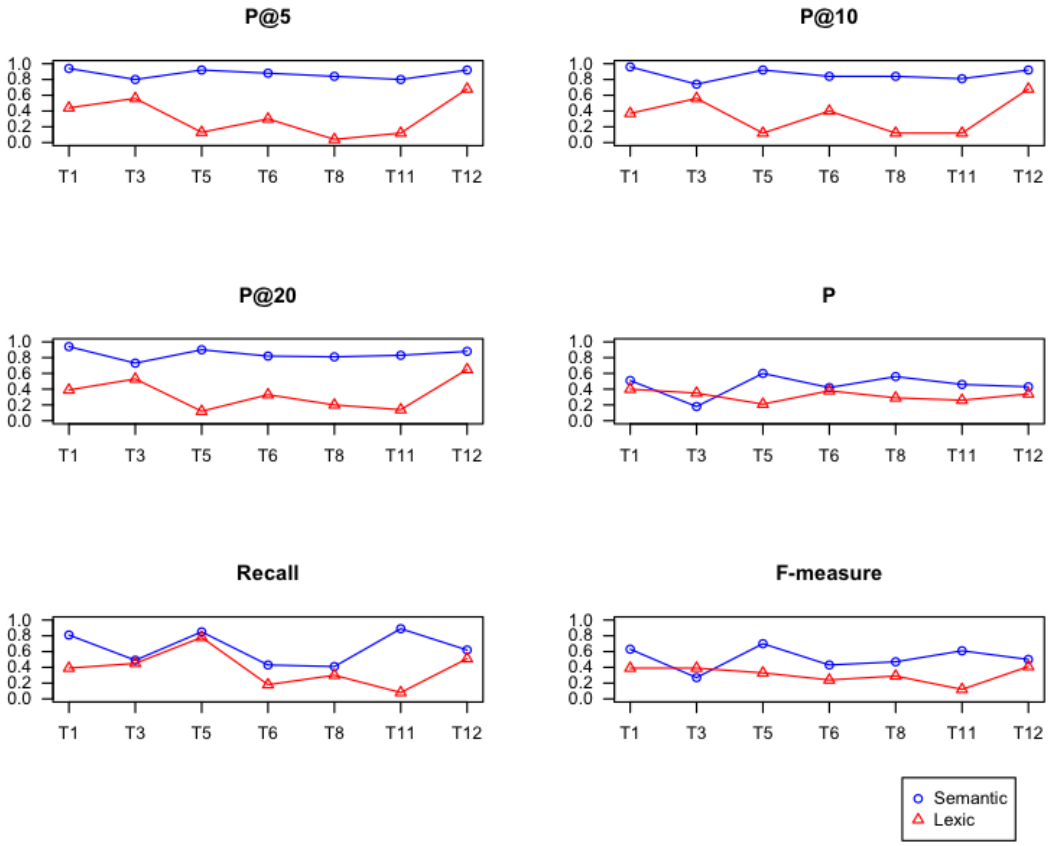

Figure 6.7: Precision of top-5, top-10, top-20, and the overall precision, recall, and F-measure of the results of the discovery using our semantic-based topic model and the results of using the keywords-based topic model.

based search and *(ii)* navigational-based search using categories. Each type of search has been evaluated separately using the GS described in Section 6.3. Next, we describe with more details each evaluation.

Keywords-based search in BioCatalogue is based on string matching techniques that consider all the available metadata of the resources. This type of search supposes an extra effort to the user, since she has to summarize her informational needs in a set of words, and these words have to make a complete matching with the words in the resource information. For instance, in BioCatalogue, the query *metabolic pathways* does not retrieve any resource, whereas its singular form *metabolic pathway* retrieves one resource. Table 6.10 shows the precision, recall, and F-measure of the results

Topic	P@5	P@10	P@20	P	R	F	Edition cost	Keywords
Search proteins with a functional domain	0.4	0.41	0.41	0.41	0.02	0.04	2.45	2.4
Search similar sequences	0.4	0.4	0.4	0.36	0.07	0.12	2.87	3.8
Analyze transgenic model organism	0.74	0.71	0.71	0.71	0.17	0.27	3.25	2.94
Find genes with functional relationships	0.27	0.26	0.27	0.26	0.04	0.07	3.15	2.13
Predict structure	0.67	0.66	0.65	0.64	0.04	0.07	3.27	2.93
Analyze phylogeny	0.18	0.2	0.2	0.18	0.01	0.02	2.8	2.56
Align sequences	0.72	0.72	0.75	0.69	0.07	0.13	2.48	4.16

Table 6.10: BioCatalogue keyword search evaluation

obtained by manually building keyword queries that try to express the informational needs described in the description tasks in the query pool. This table also shows the cost of edition, that is, the average number of failed queries we have executed before getting some results, which is in average 2.89, and the number of keywords per query, which is in average 2.94. Considering the precision and the recall, keyword queries do not provide good results considering user's requirements. Our approach presents better precision and recall without the cost of transforming the original requirements.

Navigational search allows the user to navigate through the BioCatalogue taxonomy of categories, which represent the most common bioinformatics tasks. When the user selects a category, BioCatalogue filters the resources that are tagged with that category. BioCatalogue allows to select several categories, but it does not allow to combine navigational search with keyword-based search. An important limitation of this search is that it does not retrieve uncategorized resources, even when the selected category appears in their textual description. Another limitation is the broadness of the categories, which does not allow the user to express specific tasks. To evaluate the navigational search, we have selected manually the most suitable categories for each query in the query pool. Table 6.11 shows the precision, recall and the F-measure of the results, and the cost of edition of the queries. In this type of search, the cost of edition is represented by the depth of the category in the taxonomy and the number of siblings of the selected category, describing in this way the steps required to select

Topic	P@5	P@10	P@20	P	R	F	Edition Cost
Search proteins with a functional domain	0.92	0.92	0.82	0.75	0.15	0.25	2.67/3.3
Search similar sequences	1.0	1.0	1.0	1.0	0.3	0.46	2.0/4.25
Analyze transgenic model organism	0.8	0.9	0.95	0.94	0.4	0.56	0.03/10.77
Find genes with functional relationships	0.91	0.95	0.89	0.89	0.26	0.4	1.0/3.0
Predict structure	0.87	0.93	0.96	0.9	0.1	0.18	2.29/3.42
Analyze phylogeny	0.8	0.88	0.88	0.88	0.03	0.06	0.0/11.0
Align sequences	0.98	0.99	0.99	0.99	0.06	0.11	2.94/2.1

Table 6.11: BioCatalogue navigational search evaluation

the most appropriate category. The higher the depth, the more specific the category is. As it can be noticed, the navigational search works quite well, since it also relies on semantic classification of resources, but manual. On average, the precision is high, but it is not possible to know if the retrieved results perform the specific task described in the requirement. Our approach presents a lower precision but a higher recall, that is, it retrieves relevant resources that the navigational search does not retrieve, e.g., those that are not categorized. Moreover, our approach retrieves resources that perform the specific tasks described in the requirements, which is not always possible with the navigational search due to the low specificity degree of the categories.

Table 6.8 shows a comparison between the results of the two evaluations of BioCatalogue and our approach. It can be noticed that our approach and the navigational search present similar precision, but our approach presents a higher recall.

Another important limitation of both types of search is that they do not provide a ranked list, so the user has to manually check all the results. Nevertheless, our approach provides the user with a ranked list of resources depending on their suitability to the requirement.

Regarding facets, BioCatalogue allows the user to search by introducing input or output data examples, retrieving those resources that require or produce the introduced data. However, they do not combine this search with the other types of search and, therefore, the user cannot specify which task she wants to perform over those data. In contrast, in our approach, the user can describe the required functionality together with information about the facets in the same query.

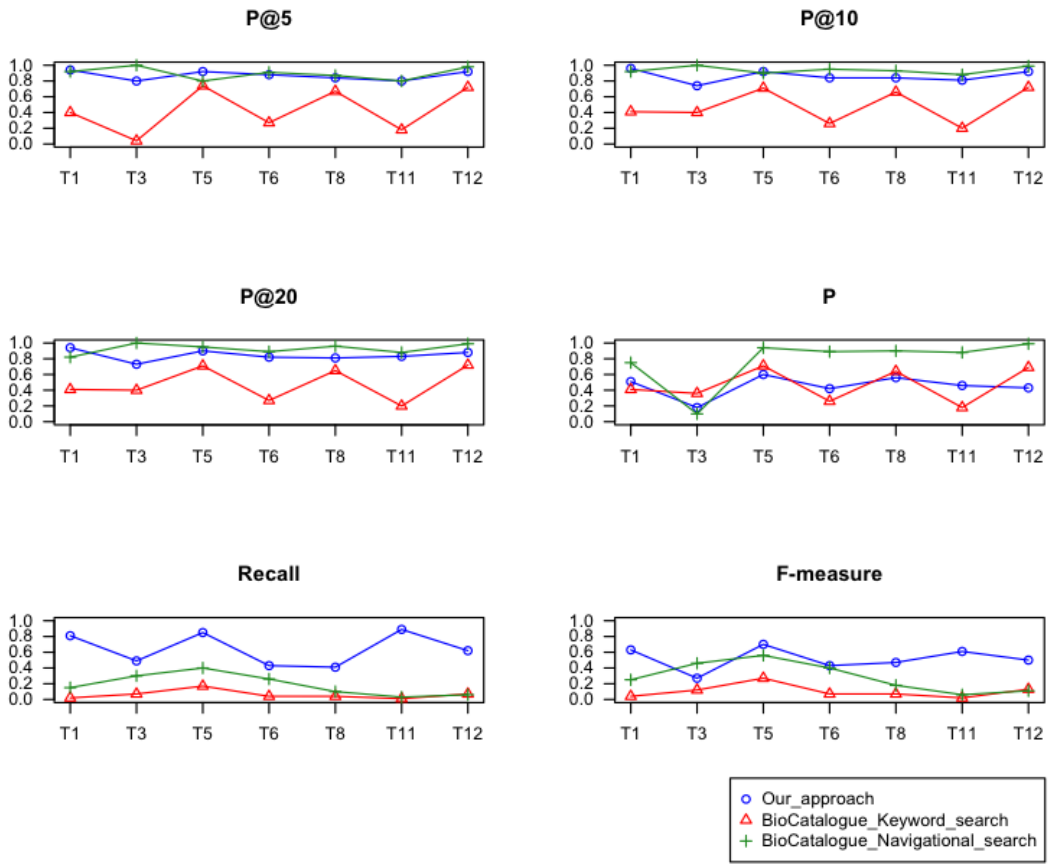

Figure 6.8: Precision of top-5, top-10, top-20, and the overall precision, recall, and F-measure of the results of our topic-based model and BioCatalogue results.

We can conclude that our approach improves the BioCatalogue search by using richer queries, which describe the task and the features of the resources, and avoiding the selection of keywords or categories that might not cover specific tasks. Moreover, the use of semantics addresses the problem of using different vocabularies or string mismatchings.

6.6 Conclusions

The results shown in this chapter have shown that the discovery approach proposed in this thesis obtains good results in the Life Sciences domain. These results demonstrate

the validity of our hypothesis stated in Section 1.2, i.e., the normalization of data overcomes the main limitations of current registries.

First, the semantic annotation of data alleviates the problem of data heterogeneity and ambiguity, and it allows to process the data automatically. The experiments reflect that the discovery based on semantics obtains much better precision, mainly in the top ranked resources, than the discovery based on the keywords. Moreover, they also show that the use of several KRs in the annotation process provides a high coverage of the different vocabularies used in Life Sciences.

Then, to characterize a resource independently of its metadata characteristics, we use a topic-based model and knowledge extraction techniques. The experiments have shown that using a topic-based model to characterize the resources metadata obtains better results than using other models, like LDA, or than not using any characterization model. On the other hand, the proposed knowledge extraction techniques identify quite well the information about five relevant features that current approaches do not consider (input, output, method, disease, and species).

The overall system obtains good results with a high precision on the top ranked resources and a high recall in average. Moreover, the proposed discovery system provides an easier and more precise way of specifying the requirements than other registries such as BioCatalogue, as demonstrated in the experiments. We think that the richer the specification of user's requirements is, the faster and the more precise the discovery of the most suitable resources is.

In conclusion, our proposed discovery system achieves a high reconcilement between user's requirements and web resources, even though when they are not described with the same vocabulary or at the same level of specificity, thanks to the normalization of data. Moreover, the system makes the discovery easier for the user, assisting her from the specification of her requirements until the selection of the most adequate web resource from a ranked list.

Chapter 7

The Prototype

The proposed web resource discovery process described in previous chapters has been implemented as part of BioUSeR, a tool for the discovery of web resources in the Life Sciences domain. The main goal of BioUSeR is to assist the user during the whole discovery process, allowing her to modify parameters in each one of the phases in order to perform a more accurate discovery.

To show the usefulness of BioUSeR, we present two example use cases that differ in the technique used to specify the user's requirements. The first example shows a case in which the requirements specification consists of a textual description, whereas in the second example the requirements are formally described with an i^* model created with the BioUSeR i^* editor. Both use cases are described by means of a sequence of BioUSeR screenshots.

Section 7.1 briefly explains the architecture of BioUSeR, and Section 7.2 shows the usefulness of the prototype by means of the two mentioned.

7.1 BioUSeR

BioUSeR is a tool for the discovery of web resources in Life Sciences whose main goal is to make the discovery of relevant resources easier for users. BioUSeR visualizes the results of the three phases of the proposed discovery process: *(i)* the user's requirements specification, *(ii)* the normalization of data, and *(iii)* the web resource discovery and ranking. Moreover, in order to gather a rich specification of the user's information needs, BioUSeR allows the user to modify the results of the normalization process and some relevant search parameters, such as the weights of the facets in the relevance

function. The simplicity of the interface, the visualization of relevant information during the whole process, and the possibility of customizing relevant parameters make the discovery of web resources easier, more intuitive, and less error-prone than in current Life Sciences registries.

BioUSeR architecture is shown in Figure 7.1. Its components are organized in three levels: *(i)* the *user interface* level, which provides a graphical interface of the discovery system for the end-user, *(ii)* the *discovery* level, which contains the core components that implement the algorithms presented in the preceding chapters, and *(iii)* the *storage* level, which stores the data sources involved in the discovery process. Next, each one of these levels is further described.

- **User interface level**. It consists of a graphical interface, implemented using GWT[1], that visualizes each one of the phases of the discovery process in three different tabs:

    1. **Requirements Specification.** This tab allows the user to provide the specification of her requirements. Currently, BioUSeR provides two ways to specify the user's requirements: a Google^{TM}-like search, in which the user can specify her requirements with a textual description, and an i^* model editor, which allows the user to create an i^* model describing the goals and tasks required to achieve her information needs.
    2. **Normalized Requirements.** This tab shows the results of the normalization of the requirements specification. It is divided into two parts: *(i)* the semantic annotation and *(ii)* the automatically identified facets values. The semantic annotation of the requirements specification is represented by pairs *concept-word*, and for each pair the user can select the ancestor of the concept to consider it in the semantic mapping performed in the discovery. The second part of the tab shows the values of the facets, which have been automatically identified in the normalization process, and their weights in the relevance function. Each facet has a default weight that can be modified by the user in order to determine the relevance of each facet in her requirements.
    3. **Resources**. This tab shows a list of the retrieved resources ranked according to their relevance to the user's requirements. Additionally, it visualizes a

[1] `https://developers.google.com/web-toolkit/`

summary of the metadata of each resource.

Once the user has specified her requirements in the *Requirements Specification* tab, the discovery process starts automatically and its results are shown in the *Normalized Requirements* and *Resources* tabs.

- **Discovery level**. It contains the core functionality of BioUSeR. It implements the process of normalization of data and the discovery of web resources in a set of Python modules. The main components are:
  - The *Normalization* component, which contains the semantic annotator and the knowledge extraction module that characterizes the data, both explained in Chapter 4.
  - The *Discovery engine*, which implements the IR model proposed in Chapter 5. It implements the discovery of web resources and their ranking according to their relevance to the user's requirements.
- **Storage level**. This level provides support for storing the resources involved in the discovery process: the knowledge resources used by the semantic annotator, and a repository of normalized metadata. In order to optimize queries during the discovery process, inverted file indexes have been used to store the information. To optimize further the discovery process, the normalization of the web resources was carried out offline. Currently, BioUSeR contains normalized metadata for 2260 resources from BioCatalogue, 2725 resources from SSWAP, and 1241 resources from myExperiment.

7.2 Example Use Cases

The current version of BioUSeR allows the user to specify her requirements by using either a Google^{TM}-like search or the i^* editor. The former allows the user to specify her requirements with a rich textual description and, from this description, BioUSeR identifies automatically relevant information about the features of the resource, which can be later modified by the user. The latter allows the user to create an i^* model describing her information needs by means of goals and tasks to achieve them. Unlike the Google^{TM}-like search, this model allows defining more than one task, and also dependencies between them (e.g., the output of a task can be the input of the following

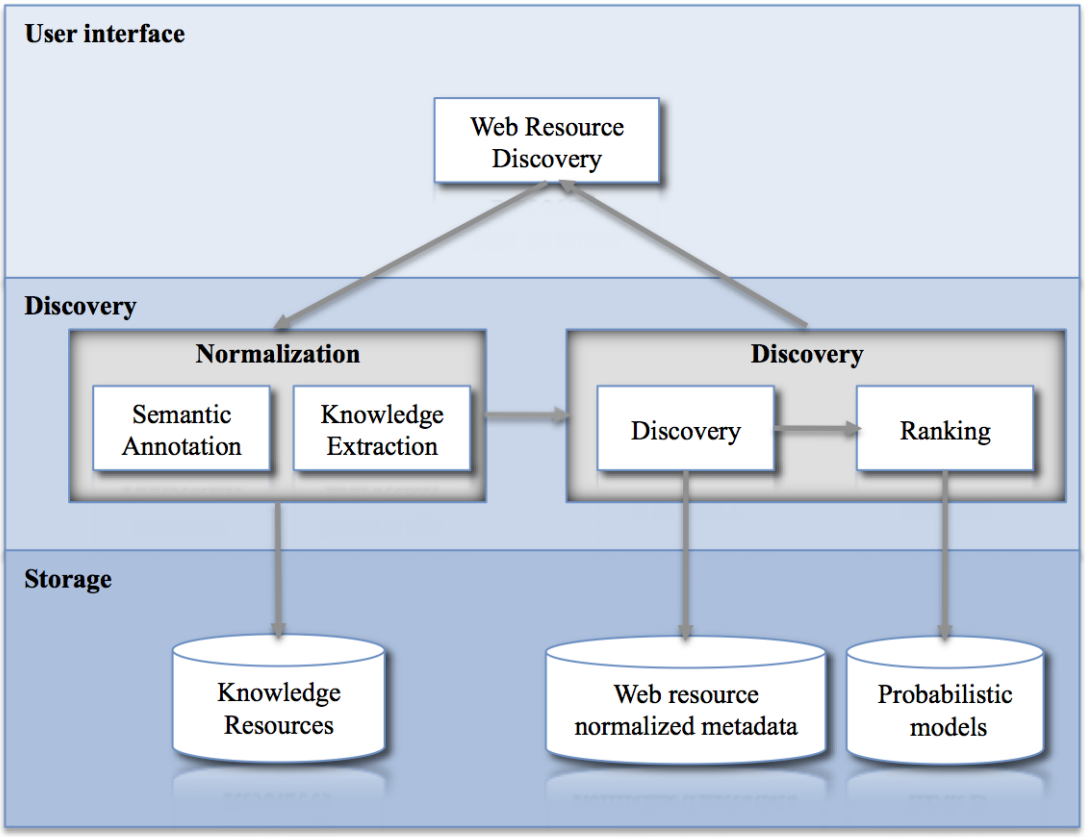

Figure 7.1: Architecture of BioUSeR

task). Although currently BioUSeR only supports these two techniques, other requirements specification techniques could be supported by implementing their own extractor module, as explained in Section 3.1.

In this section, we present two example use cases. Section 7.2.1 presents a case in which the user's requirement is described with a single textual description, and Section 7.2.2 shows a case in which the user's requirement is specified by an i^* model.

7.2.1 Discovery driven by a Textual Description

In current registries, the input of the discovery is specified in a text box in which the user writes the keywords that best represent her requirements. BioUSeR also

provides this type of search, but without the restriction of using specific vocabularies or specific formats. Therefore, the user can describe her information needs with a textual description that describes not only the required functionality, but also relevant features of the resources. In this section, we present an example of this type of search to show how discovery driven by a single textual description is performed in BioUSeR.

The use case is about a user that needs to align sequences given in Fasta format and using the Smith-Waterman algorithm. Figure 7.2 shows a screenshot of the requirement specification *Align sequences using Smith-Waterman given Fasta.*

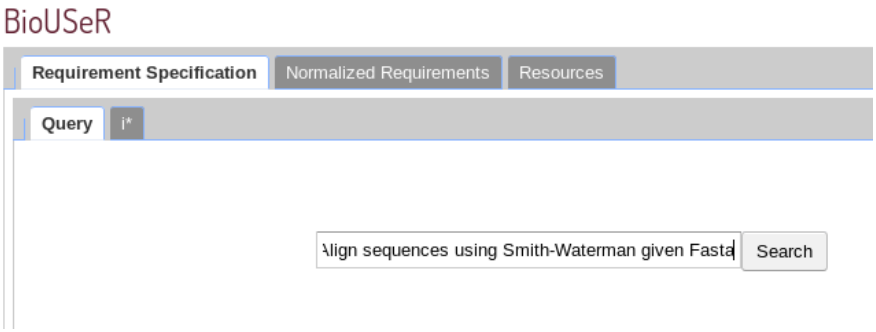

Figure 7.2: Requirement specification using a textual description.

Once the user has specified the textual description of her requirement, the results of the normalization process are automatically shown in the *Normalized Requirements* tab, as shown in Figure 7.3. In the part of the semantic annotation, BioUSeR shows the concepts associated to each entity in the requirements specification, e.g., the word *Fasta* has been annotated with the concepts W1009996, D9000079, and C1708003. In case the user wants to use a more general concept, she can select an ancestor concept in order to be considered in the discovery. For example, in Figure 7.3 it can be shown that the ancestor of the concept C1708003*(fasta)* is C1301627 *(format)*. In the part of the facets, BioUSeR shows the facets values that have been automatically identified and their corresponding weights in the relevance function. In this example usage case, BioUSeR has identified *Fasta* as input value, and *Align multiple sequences using Smith-Waterman* as method.

Finally, Figure 7.4 shows the *Discovered resources* tab that presents the user the top-10 resources that are supposed to be the ones that best fulfill the user's requirement. In addition, a summary of the metadata of each resource is shown.

If the user does not get the expected resources, she can refine the discovery by

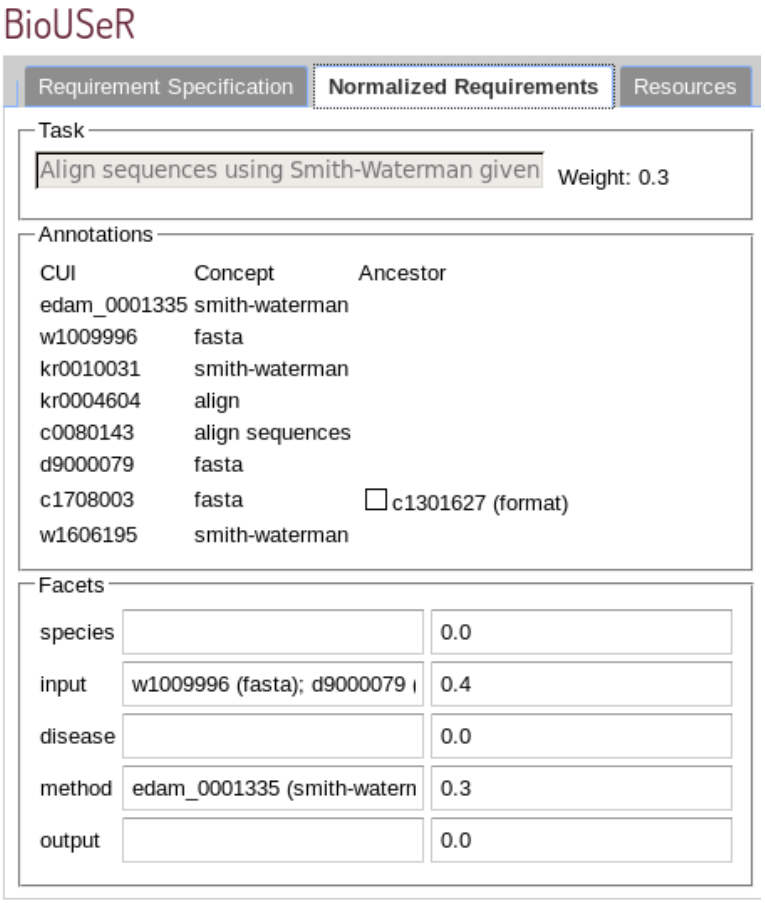

Figure 7.3: Normalization of the user's requirement specification.

modifying the textual description, selecting more general concepts with the ancestors or modifying the facets values or their weights in the relevance function. For example, if the user wants to give more relevance to the facet *method*, the user can modify the weight of that facet. Figure 7.5 shows that the weight of the facet *method* has been increased from 0.3 to 0.5. Figure 7.6 shows the new ranked list of resources in which the resources performing Smith-Waterman as alignment method are now top-ranked.

7.2.2 Discovery driven by an i^* Model

This section presents an example use case in which the user's requirement is described by means of an i^* model, composed by goals and tasks that must be performed by web resources in order to achieve them.

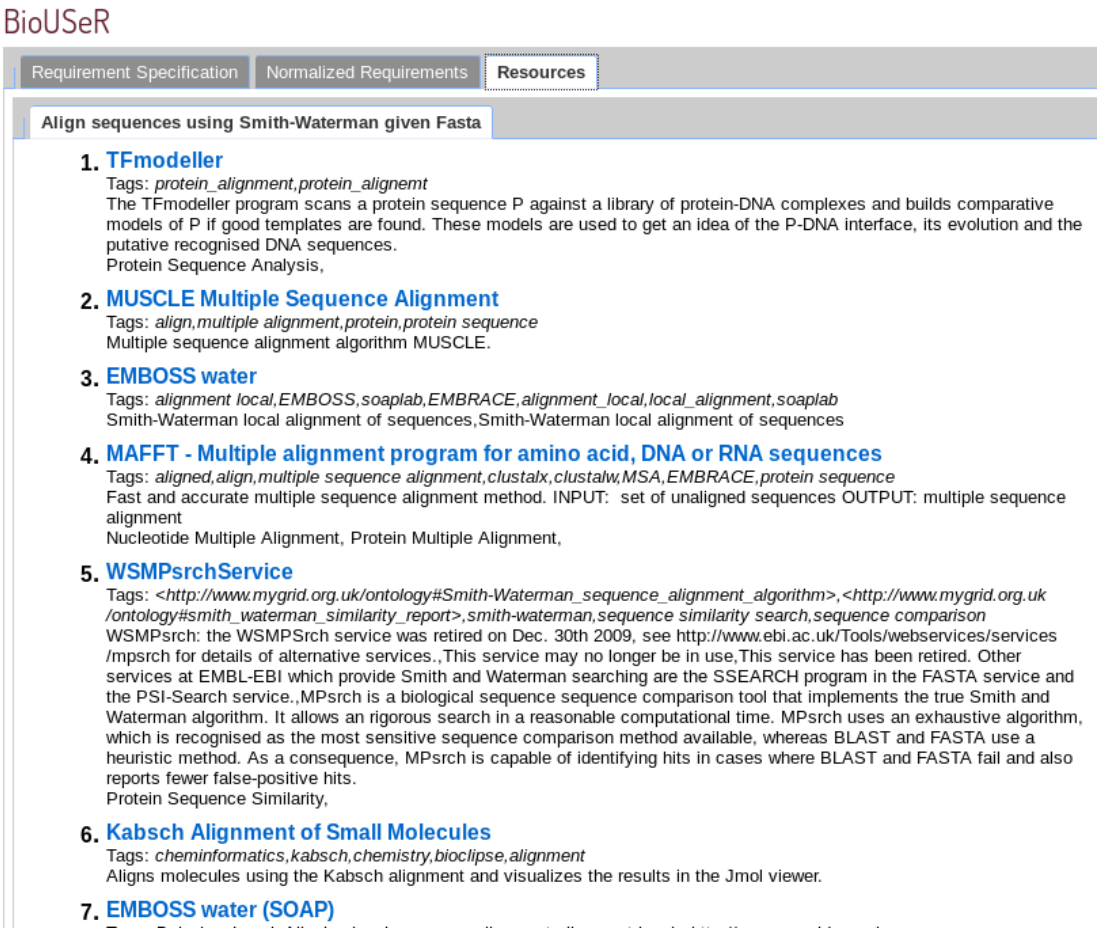

Figure 7.4: Results of the web resource discovery process.

In this use case, extracted from [59], the user needs to compare specific genes in different organisms. Figure 7.7 shows the i^* model that describes the user's requirement, in which the tasks that must be performed by web resources are: *Search similar sequences given a protein sequence*, *Predict gene structure*, *Align protein sequences*, *Build phylogenetic trees*, and *Analyze domains given protein sequences*.

Then, the semantic annotation and the automatically identified facets values of each task in the i^* model are shown in the *Normalized Requirements* tab, as shown in Figure 7.8. Figure 7.9 shows the normalization of the tasks *Search similar sequences given a protein sequence* and *Predict gene structure*. As shown in the figure, the output of the first task is the input of the second task, due to the dependencies between tasks in the

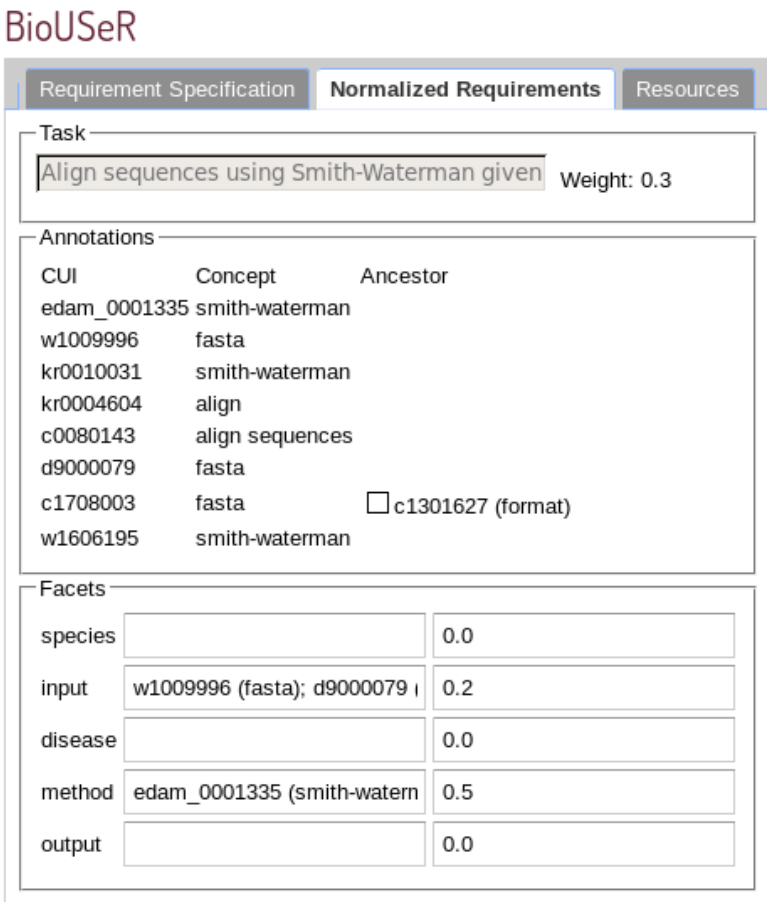

Figure 7.5: Modification of the weight of the facet *method*.

i^* model.

Finally, the discovered resources for each of the user-defined tasks are visualized in the *Resources* tab. Figure 7.10 shows the discovered resources for the task *Predict gene structure*.

7.3 Conclusions

The tool presented in this chapter demonstrates the usefulness of the techniques proposed in this thesis. Its aim is to make the discovery of web resources easier for researchers who are looking for resources that fulfill their information needs. BioUSeR allows providing a rich description of the user's information needs, not only by the

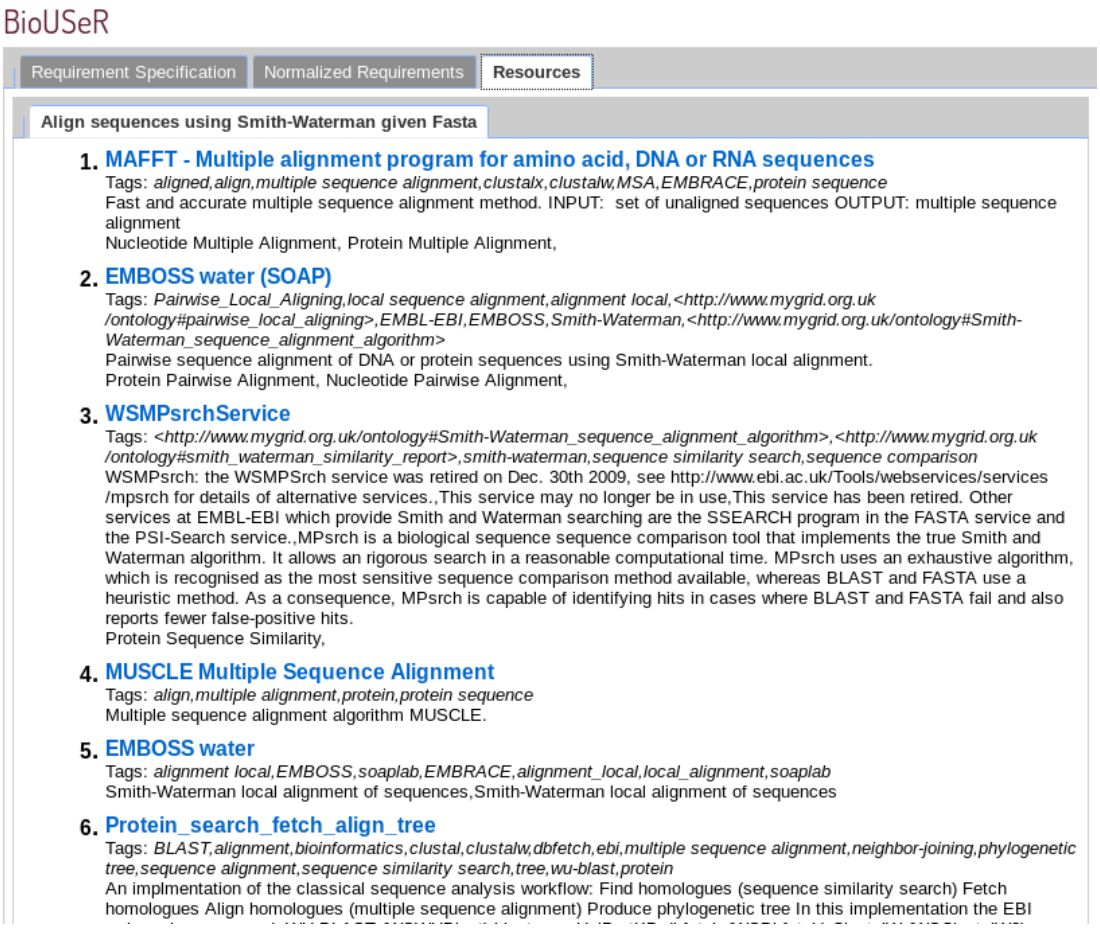

Figure 7.6: New results after modifying the weight of the facet *method*.

initial requirements specification, but also by the possibility of customizing search parameters. Moreover, with the aim of being widely adopted by different types of users, BioUSeR supports different techniques to specify the user's requirements.

The visualization of relevant information involved in the discovery process and the possibility of customizing this information help users find the most relevant resources faster, and with less effort, than in current Life Sciences registries.

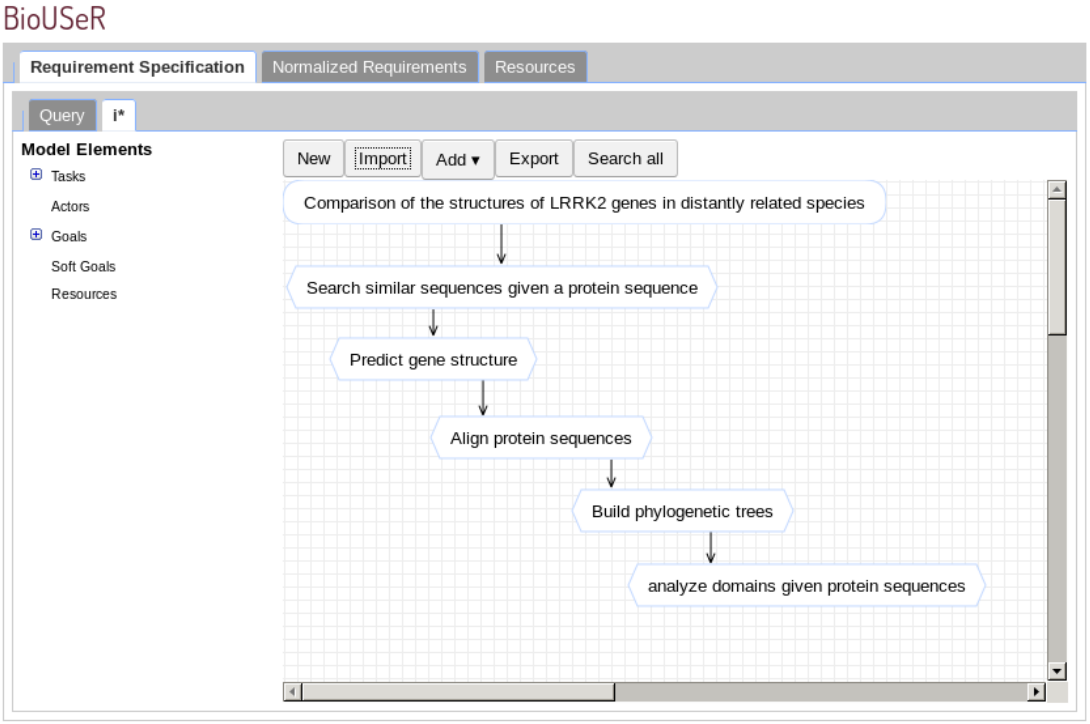

Figure 7.7: Requirements specification using an i^* model.

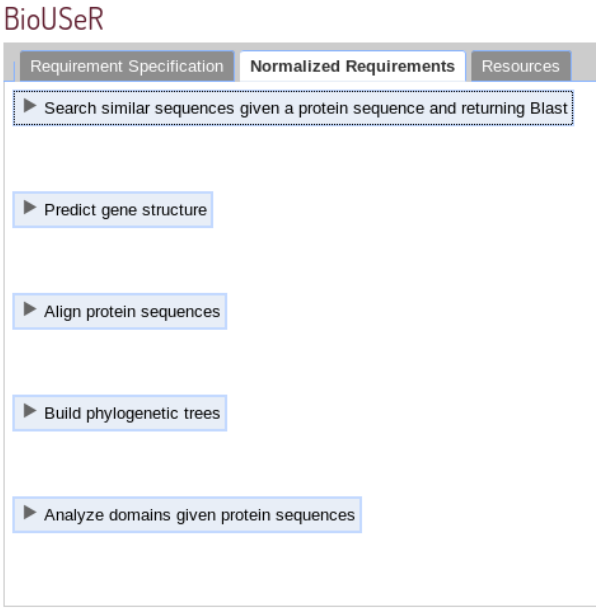

Figure 7.8: Normalization of the user's requirements specification.

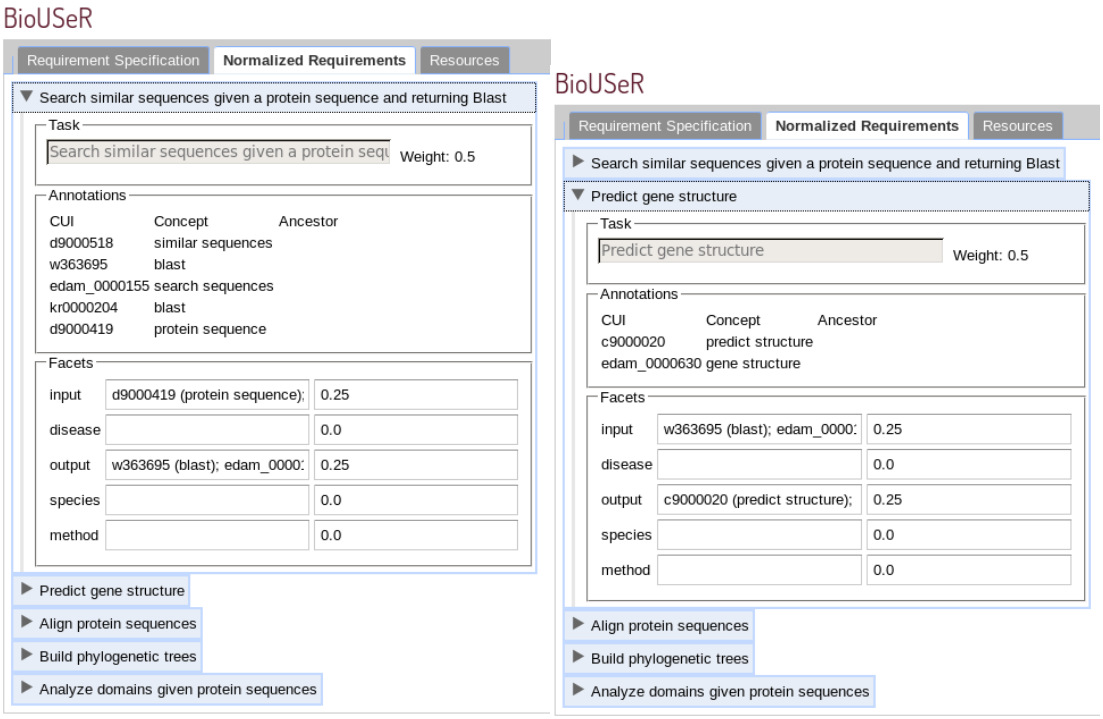

Figure 7.9: Normalization of the tasks *Search similar sequences given a protein sequence* and *Predict gene structure*.

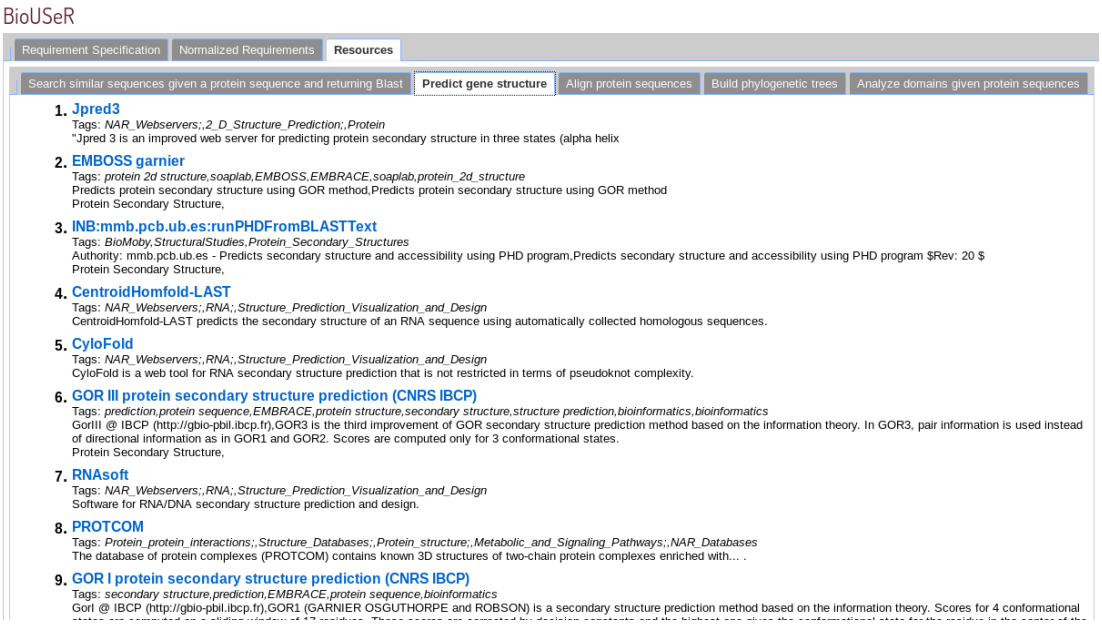

Figure 7.10: Results of the web resource discovery process for the task *Predict gene structure*.

Chapter 8

Conclusions

This last chapter summarizes the main results of the thesis and outlines possible future research lines. The chapter concludes by listing the publications resulted from this thesis work. Section 8.1 surveys the results of the thesis. Section 8.2 discusses the future work. Finally, Section 8.3 lists the main published contributions of the thesis.

8.1 Summary of Results

In last years, the number of web resources available on the Web has increased at a vertiginous rate. In Life Sciences, researchers are publishing their research results (e.g., data sets and processing tools) on the Web with the aim of collaborating with other researchers. However, the discovery of web resources relevant to a specific requirement is a challenge task for Life Sciences researchers, due to the huge amount of web resources, their high heterogeneity, and the lack of adequate metadata describing them.

This thesis has reviewed the discovery of web resources in the Life Science domain. Currently, there are web resource registries on the Web that allow users to discover web resources that are supposed to be relevant to their requirements. However, most of them present some limitations that hinder the web resource discovery: *(i)* poor representation of user's requirements, *(ii)* high discovery dependency on the characteristics of the resources metadata, and *(iii)* low assistance to the user during the whole discovery process, specifically in the selection of the most appropriate resource.

The goal of this thesis is to assist the user in the discovery of the resources that are the most appropriate for her requirements, by addressing the main limitations of current registries. The main characteristic of the proposed approach is that the whole

discovery process is driven by the user, who is assumed to provide a rich specification of her requirements, and who can modify the discovery parameters in order to customize the process.

In the proposed approach, the specification of the user's requirements is a rich description of what the user needs, which includes not only the functionality, but also relevant features of the required resource, such as the input/output parameters, the species, and the diseases involved by the resource. With the aim of being widely adopted by users, our approach is not restricted to a specific requirement specification technique. A priori, any technique can be supported by developing a specific information extraction module. Currently, the implemented prototype, BioUSeR, supports textual descriptions and i^* models as requirements specification techniques.

One of the most important limitations of current registries is their high dependency on the characteristics of metadata, both structural and lexical. With respect to the structural dependency, many registries define specific fields to describe specific features of the resources. However, evidence shows that most of the resources features are implicitly described in the textual descriptions and, in consequence, they are not identified as feature values by the search engines. On the other hand, regarding the lexical dependency, the lack of widely accepted standards increases the heterogeneity of data describing the resources and, therefore, users have to know which vocabulary has been used in the metadata in order to specify their requirements with the same vocabulary. The discovery process proposed in this thesis alleviates this dependency by using normalization techniques. First, to address the heterogeneity and ambiguity of data, all data involved in the discovery process are semantically annotated with domain knowledge resources. Afterwards, knowledge extraction techniques are used to automatically identify relevant information about the resources features, which improve their characterization. Then, the discovery is based on the semantic mapping of the normalized requirements specification and the normalized resources metadata, retrieving in this way resources described with different styles and vocabularies. Therefore, we can conclude that the dependency on the characteristics of the metadata is considerably reduced by the use of normalized data.

With the aim of assisting the user until the end of the discovery process, the discovered resources are ranked according to their relevance to the user's requirement. The relevance of a resource is estimated considering how well the resource fulfils not only the functionality, but also the features required by the user. At the end, the user gets a

ranked list in which the most appropriate resources for her requirements are in the top positions. Finally, if the resources are not those expected by the user, she can modify the discovery process by modifying the requirements specification, the information about the facets automatically identified, and other discovery parameters.

Therefore, we can conclude that the main limitations of current registries in Life Sciences have been alleviated in our approach by: *(i)* allowing the user to provide a rich specification of her information needs and to modify discovery parameters and information that have been automatically identified (e.g., facets values), *(ii)* using normalization techniques in order to alleviate the dependency on the data characteristics, and *(iii)* providing relevant information to the user, such as the automatically extracted facets values, the semantic annotation of her requirements specification, and the ranking of resources.

The discovery approach has been validated by evaluating each one of its phases. Moreover, we have further validated it by comparing it with other IR techniques, and with one of the most popular web resource registries in Life Sciences, BioCatalogue. This later experiment has demonstrated that our approach obtains more precise results with less iterations and fewer effort than current registries.

Finally, the proposed discovery process has been implemented as part of a prototype called BioUSeR. BioUSeR visualizes each phase of the discovery process, and allows the user to modify some parameters during the whole process. Its simplicity and the visualization of relevant information make the discovery of relevant web resources less hard and less error-prone.

8.2 Future Work

A number of directions for further research have been pointed out throughout the thesis, which we summarize here. First, we point out to specific limitations of the current developed methods and suggest further improvements. Then, we refer to more general research lines that have emerged from this thesis.

With respect to the normalization of data, the coverage and precision of the semantic annotation of data can be improved in several aspects. Currently, the knowledge resources are considered independent of each other, when they actually share some concepts. To reduce redundancy in the semantic annotations, the knowledge resources should be aligned in order to associate equivalent concepts. Therefore, the knowledge

resources alignment would reduce the number of concepts in the semantic annotations and their ambiguity. Moreover, new subsumption relationships between concepts of different KRs could be automatically identified from these alignments.

Another aspect that could be improved in the semantic annotation process is the post-processing of the annotations. Currently, the simplification of annotations is made based on the specificity of the knowledge resources and the specificity of the concepts, given by their *idf* score. However, ambiguous annotations are not completely disambiguated with these simplification techniques. We think that the disambiguation techniques that consider the context of the annotation could select the concepts that give the correct sense of the annotation. Recently, we have made some preliminary work on context-aware disambiguation techniques, presented in [48], whose results are encouraging.

Regarding the IR model, it can be improved by having more information under consideration in the ranking process of the discovered resources. There are several types of information which can be relevant to get a more precise ranking, in which the top-ranked resources are the most adequate for the user's requirement.

Firstly, user's feedback and results of similar cases provide relevant information about the accuracy of the discovery method. We believe that considering feedback in the process of ranking of the discovered resources may obtain more precise results in the top-ranked positions.

A second way to improve the ranking of resources is the consideration of non-functional requirements (NFRs). The most popular NFRs describe features about the resource performance and users' opinions. Currently, few registries show information about the performance of the resources (e.g., EMBRACE), whereas the most recent registries show information about users' opinion by means of comments and ratings (e.g., BioCatalogue and myExperiment). Although NFRs are not relevant during the discovery process, they can improve the ranking of resources when the user considers them an important criterion in the selection of a resource. In our approach we do not consider NFRs, but we think that it could be interesting to support them, specifically social NFRs, in order to provide further information to the user. These NFRs could be considered in the ranking of resources and visualized together with the resources metadata.

Finally, the ranking of resources could also be more accurate by considering the context of the requirement, when available. For example, BioUSeR allows users to

create i^* models in which the user can specify dependencies between tasks. Now, the only processing of these dependencies consists in assigning as input of a task the output of the previous task. It would be very interesting to reorder the ranking of resources considering the resource selected for the previous task, with the aim of improving the interoperability between resources. So, when a user selects a resource for a specific task, the ranked list of resources of the next task would be re-ranked accordingly. This technique could be applied to any requirements specification format in which dependencies between tasks are explicitly defined (e.g., graphs).

With a wider perspective, the techniques proposed in this thesis can be used in any application that retrieves, integrates or compares resources. Although this thesis is focused on the Life Sciences domain, the proposed approach could be easily adapted to other domains, by selecting adequate KRs, and creating the corresponding models. In general, these techniques are aimed to retrieve resources that are annotated with few well-defined metadata, but with rich textual descriptions.

For example, the proposed techniques could be applied on storage systems that contain different types of resources (e.g., images and reports) with different characteristics and different metadata. Given a specific user's requirement, our approach would discover different types of resources, however they are described, but relevant to the user's requirement. Moreover, similarity search could be also possible given a specific resource thanks to the use of semantics. Therefore, our approach would make the discovery and integration of information coming from different types of resources easier for the users.

Another application example are current catalogues of applications such as App Store, which base the discovery on well-defined features and the string matching in the resources descriptions. We believe that our techniques would improve the discovery of the resources in those catalogues, by allowing the users to define their requirements with richer textual descriptions, without worrying about which vocabulary to use and in which field to search.

Another interesting research line is workflow composition. Currently, the composition of workflows is limited by the availability of well-defined metadata of web resources, in concrete, the input/output data types. Our normalization techniques would be very useful to identify such features, even when they are explicitly described. Moreover, using requirements specification formats which allow users to define dependencies between tasks would make the composition of workflows possible in our approach.

In conclusion, the techniques proposed in this thesis can be used to discover web resources, whatever their type and how they are described, and to facilitate the integration of data.

8.3 List of Publications

This section enumerates the publications derived from this thesis, grouped by topics and pointed out the chapters that mainly influenced them.

The rich specification of user's requirements was first addressed in [80], in which *i** framework was used to design similarity measures. This requirements specification technique has been later used in the rest of publications.

- [80] *María Pérez*, Sven Casteleyn, Ismael Sanz and María José Aramburu. Requirements gathering in a model-based approach for the design of multi-similarity systems. In Proceedings of *MoSE+DQS'09 Workshop in Conference on Information and Knowledge Management (CIKM'09)* .

The work done about the normalization of data has been presented in several conferences. The work in [83] and [76] describes briefly the discovery of resources based on semantic annotations.

- [83] *María Pérez*, Ismael Sanz, Rafael Berlanga and María José Aramburu. Adding Semantics to the Discovery of Web Services. In Proceedings of *9th International Conference on Practical Applications of Agents and Multi-Agent Systems*, volume 90, pages 145-152, 2011.

- [76] *María Pérez*, Rafael Berlanga and Ismael Sanz. A semantic approach for the requirement-driven discovery of web services in the Life Sciences. In Proceedings of *3rd International Workshop on Semantic Web Applications and Tools for Life Sciences (SWAT4LS 2010).*

With respect to the characterization of resources, the early work in [77] presents a faceted search in which resources facets are extracted using patterns. This paper demonstrates that the faceted search improves the specification of user's requirements and, consequently, the discovery results. To not depend on a set of specific patterns, which implies a dependency on the vocabulary used in the resources metadata, [81]

describes the extraction of facets using a translation model, which is the foundation of the techniques developed in Chapter 4.

- [77] *María Pérez*, Rafael Berlanga, Ismael Sanz and María José Aramburu. Exploiting text-rich descriptions for faceted discovery of web resources. In Proceedings of *4th International Workshop on Semantic Web Applications and Tools for Life Sciences (SWAT4LS 2011).*

- [81] *María Pérez*, Lisette García-Moya and Rafael Berlanga. A translation model for facet-based retrieval in open registries. In Proceedings of *II Congreso Español de Recuperación de Información (CERI 2012)* .

As future work, we have mentioned in last section that we aim to improve the post-processing of semantic annotations by using context-aware techniques to disambiguate semantic annotations. Some work has been already done on this line. [48] presents a disambiguation method based on the similarity between concept profiles, generated from Medline, and the contexts in which the concepts appear.

- [48] Antonio Jimeno-Yepes, *María Pérez* and Rafael Berlanga. Disambiguating automatically-generated semantic annotations for Life Sciences open registries. In Proceedings of the *2nd International Workshop on Exploiting Large Knowledge Repositories (E-LKR'12).* Castellón (Spain), September 2012.

With respect to the discovery process, the main idea was first presented in [84]. The most important publications about the whole discovery process are the journal articles [79] and [78], in which the whole discovery process is described with detail.

- [84] *María Pérez*, Ismael Sanz, Rafael Berlanga, María José Aramburu. Semi-automatic discovery of web services driven by user requirements. In Proceedings of *21st International Conference on Databases and Expert Systems Applications (DEXA 2010)*, volume 6261, pages 62-75, 2010.

- [78] *María Pérez*, Rafael Berlanga, Ismael Sanz and María José Aramburu. A semantic approach for the requirement-driven discovery of web resources in the Life Sciences. In *Knowledge and Information Systems*, 34:671-690, 2013.

- [79] *María Pérez*, Rafael Berlanga, Ismael Sanz and María José Aramburu. BioUSeR: A semantic-based tool for retrieving Life Sciences resources driven by text-rich user requirements. In *Journal of Biomedical Semantics*, 4:12, 2013.

Also as a future research line, we aim to provide further information to the user in order to assist her in the selection of the most appropriate resource. As a possible technique for giving assistance to the user, [82] describes a biclustering-based technique to find similarities between the discovered resources. The aim of this technique is to provide the user with groups of resources that present similar features and, therefore, they can be considered equivalent.

- [82] *María Pérez*, Ismael Sanz and Rafael Berlanga. A biclustering-based technique for requirement-driven Web Service selection. In Proceedings of *XV Jornadas de Ingeniería del Software y Bases de Datos (JISBD 2010)*

Bibliography

[1] Hammad Afzal, James Eales, Robert Stevens, and Goran Nenadic. Mining semantic networks of bioinformatics e-resources from the literature. *Journal of Biomedical Semantics*, 2:S4, 2011. 13

[2] Hammad Afzal, Robert Stevens, and Goran Nenadic. Towards semantic annotation of bioinformatics services: Building a controlled vocabulary. In *Proceedings of the Third International Symposium on Semantic Mining in Biomedicine (SMBM 2008)*, pages 5–12, 2008. 44

[3] Eyhab Al-Masri and Qusay H. Mahmoud. Discovering Web Services in Search Engines. *Internet Computing, IEEE*, 12(3):74 –77, 2008. 13

[4] Eyhab Al-Masri and Qusay H. Mahmoud. WSB: a broker-centric framework for quality-driven web service discovery. *Software: Practice and Experience*, 40(10):917–941, 2010. 18

[5] Omar Alonso and Hugo Zaragoza. Semantic Annotations in Information Retrieval. *Information Processing & Management*, 46, Issue 4:381–494, July 2010. 2

[6] Alan R. Aronson. Effective mapping of biomedical text to the UMLS metathesaurus: The MetaMap program. In *Proceedings of the 2001 AMIA Symposium*, pages 17–21, 2001. 19

[7] Franz Baader, Diego Calvanese, Deborah McGuiness, Daniele Nardi, and Peter Patel-Schneider. *The Description Logic Handbook: Theory, Implementation, and Applications*. Cambridge University Press, 2003. 15

[8] Ricardo A. Baeza-Yates and Berthier Ribeiro-Neto. *Modern Information Retrieval*. Addison-Wesley Longman Publishing Co., Inc., 1999. 2, 10

[9] Luciano Barbosa and Juliana Freire. Combining classifiers to identify online databases. In *Proceedings of the 16th international conference on World Wide Web*, WWW '07, pages 431–440, New York, NY, USA, 2007. ACM. 19

[10] Robert Baumgartner, Sergio Flesca, and Georg Gottlob. Visual Web Information Extraction with Lixto. In *Proceedings of the 27th International Conference on Very Large Data Bases*, VLDB '01, pages 119–128. Morgan Kaufmann Publishers Inc., 2001. 16

[11] François Belleau, Marc-Alexandre Nolin, Nicole Tourigny, Philippe Rigault, and Jean Morissette. Bio2RDF: Towards a mashup to build bioinformatics knowledge systems. *Journal of Biomedical Informatics*, 41(5):706–716, 2008. 19, 24

[12] Adam Berger and John Lafferty. Information retrieval as statistical translation. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and Development in Information Retrieval*, SIGIR '99, pages 222–229. ACM, 1999. 72

[13] Rafael Berlanga, Victoria Nebot, and Ernesto Jimenez. Semantic annotation of biomedical texts through concept retrieval. In *BioSEPLN 2010*, volume 45, pages 247–250, 2010. 46, 48

[14] Tim Berners-Lee, James Hendler, and Ora Lassila. The Semantic Web. *Scientific American*, May 17 2001. 2

[15] Jiten Bhagat, Franck Tanoh, Eric Nzuobontane, Thomas Laurent, Jerzy Orlowski, Marco Roos, Katy Wolstencroft, Sergejs Aleksejevs, Robert Stevens, Steve Pettifer, Rodrigo Lopez, and Carole A. Globe. BioCatalogue: a universal catalogue of web services for the life sciences. *NAR*, 2010. 3, 15, 19, 25

[16] István Bíró and Jácint Szabó. Latent dirichlet allocation for automatic document categorization. In *Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases: Part II*, ECML PKDD '09, pages 430–441, Berlin, Heidelberg, 2009. Springer-Verlag. 73

[17] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent Dirichlet Allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003. 54, 57, 73

[18] Michelle D. Brazas, David S. Yim, Joseph T. Yamada, and B. F. Francis Ouellette. The 2011 Bioinformatics Links Directory update: more resources, tools and databases and features to empower the bioinformatics community. *Nucleic Acids Research*, 39(suppl 2):W3–W7, 2011. 19, 28

[19] Wray Buntine, Jaakko Lofstrom, Jukka Perkio, Sami Perttu, Vladimir Poroshin, Tomi Silander, Henry Tirri, Antti Tuominen, and Ville Tuulos. A Scalable Topic-Based Open Source Search Engine. In *Proceedings of the 2004 IEEE/WIC/ACM International Conference on Web Intelligence*, WI '04, pages 228–234. IEEE Computer Society, 2004. 72

[20] Ben Carterette and Praveen Chandar. Probabilistic Models of Novel Document Rankings for Faceted Topic Retrieval. In *CIKM'09*, 2009. 57, 73, 74

[21] Ran Chen, Hongfei Lin, and Zhihao Yang. Passage retrieval based hidden knowledge discovery from biomedical literature. *Expert Syst. Appl.*, 38(8):9958–9964, August 2011. 74

[22] Paolo Ciccarese, Marco Ocana, Leyla Garcia Castro, Sudeshna Das, and Tim Clark. An open annotation ontology for science on web 3.0. *Journal of Biomedical Semantics*, 2(Suppl 2):S4+, 2011. 20

[23] Philipp Cimiano, Siegfried Handschuh, and Steffen Staab. Towards the self-annotating web. In *Proceedings of the 13th International conference on World Wide Web*, WWW '04, pages 462–471. ACM, 2004. 16

[24] Fabio Ciravegna, Alexiei Dingli, Daniela Petrelli, and Yorick Wilks. User-system cooperation in document annotation based on information extraction. In *In Proceedings of the 13th International Conference on Knowledge Engineering and Knowledge Management, EKAW02*, pages 122–137. Springer Verlag, 2002. 16

[25] Francisco M. Couto, Mário J. Silva, and Pedro Coutinho. Finding genomic ontology terms in text using evidence content. *BMC Bioinformatics*, 6(S-1), 2005. 47

[26] Marco Crasso, Alejandro Zunino, and Marcelo Campo. Easy web service discovery: A query-by-example approach. *Science of Computer Programming*, 71(2):144–164, 2008. 15

[27] Wisam Dakka and Panagiotis G. Ipeirotis. Automatic Extraction of Useful Facet Hierarchies from Text Databases. In *Proceedings of the 2008 IEEE 24th International Conference on Data Engineering*, ICDE '08, pages 466–475, Washington, DC, USA, 2008. IEEE Computer Society. 58

[28] Scott Deerwester, Susan T. Dumais, George W. Furnas, Thomas K. Landauer, and Richard Harshman. Indexing by Latent Semantic Analysis. *Journal of the American Society for Information Science*, 41(6):391–407, 1990. 17

[29] Vikas Deora, Jianhua Shao, Gareth Shercliff, Patrick J. Stockreisser, W. Alex Gray, and Nick J. Fiddian. Incorporating QoS Specifications in Service Discovery. pages 252–263. 2004. 18

[30] Marie Dominique Devignes, Philippe Franiatte, Nizar Messai, Amedeo Napoli, and Malika Smail-Tabbone. BioRegistry: automatic extraction of metadata for biological database retrieval and discovery. In *Proceedings of the 10th International Conference on Information Integration and Web-based Applications & Services*, iiWAS '08, pages 456–461. ACM, 2008. 2, 19, 21

[31] Stephen Dill, Nadav Eiron, David Gibson, Daniel Gruhl, R. Guha, Anant Jhingran, Tapas Kanungo, Kevin S. Mccurley, Sridhar Rajagopalan, Andrew Tomkins, John A. Tomlin, and Jason Y. Zien. A Case for Automated Large Scale Semantic Annotations. *Journal of Web Semantics*, 1:115–132, 2003. 16

[32] Oren Etzioni, Michael Cafarella, Doug Downey, Ana-Maria Popescu, Tal Shaked, Stephen Soderland, Daniel S. Weld, and Alexander Yates. Unsupervised Named-Entity Extraction from the Web: An Experimental Study. *Artificial Intelligence*, 165:91–134, 2005. 16

[33] Kambiz Frounchi, Partheeban Chandrasekaran, Jawid Ibrahimi, Shikharesh Majumdar, Chung-Horng Lung, and Laura Serghi. A QoS-Aware Web Service Replica Selection Framework for an Extranet. In *Electrical and Computer Engineering, 2006. CCECE '06.*, pages 1380 –1384, 2006. 12

[34] María Jesús García Godoy, Esteban López-Camacho, Ismael Navas-Delgado, and José F. Aldana-Montes. Sharing and Executing Linked Data Queries in a Collaborative Environment. *Bioinformatics*, 2013. 20, 36

[35] Lisette García-Moya, Henry Anaya-Sánchez, and Rafael Berlanga-Llavori. Combining probabilistic language models for aspect-based sentiment retrieval. In *Proceedings of the 34th European conference on Advances in Information Retrieval*, ECIR'12, pages 561–564, Berlin, Heidelberg, 2012. Springer-Verlag. 61

[36] Damian DG Gessler, Gary S Schiltz, Greg D May, Shulamit Avraham, Christopher D Town, David Grant, and Rex T Nelson. SSWAP: A Simple Semantic Web Architecture and Protocol for semantic web services. *BMC Bioinformatics*, 10:309, 2009. 19, 20, 27, 28, 35

[37] Carole Goble, Robert Stevens, Ducan Hull, Katy Wolstencroft, and Rodrigo Lopez. Data curation + process curation = data integration + science. *Briefings in Bioinformatics*, 9(6):506–517, 2008. 19

[38] Carole A. Goble, Jiten Bhagat, Sergejs Aleksejevs, Don Cruickshank, Danius Michaelides, David Newman, Mark Borkum, Sean Bechhofer, Marco Roos, Peter Li, and David De Roure. myExperiment: a repository and social network for the sharing of bioinformatics workflows. *Nucleic Acids Research*, 38(suppl 2):W677–W682, 2010. 3, 19, 20, 25

[39] Karthik Gomadam, Ajith Ranabahu, Meenakshi Nagarajan, Amit P. Sheth, and Kunal Verma. A Faceted Classification Based Approach to Search and Rank Web APIs. In *IEEE International Conference on Web Services, 2008. ICWS '08.*, pages 177 –184, 2008. 56

[40] Thomas L. Griffiths and Mark Steyvers. Finding scientific topics. *Proceedings of the National Academy of Sciences of the United States of America*, 101(Suppl 1):5228–5235, 2004. 54

[41] Shengbo Guo and Scott Sanner. Probabilistic latent maximal marginal relevance. In *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '10, pages 833–834, New York, NY, USA, 2010. ACM. 73

[42] Rafael Guzmán-Cabrera, Manuel Montes-Y-Gómez, Paolo Rosso, and Luis Villaseñor Pineda. Using the web as corpus for self-training text categorization. *Inf. Retr.*, 12(3):400–415, June 2009. 54

[43] Siegfried Handschuh, Steffen Staab, and Rudi Studer. Leveraging Metadata Creation for the Semantic Web with CREAM. In Andreas Gnter, Rudolf Kruse, and Bernd Neumann, editors, *KI 2003: Advances in Artificial Intelligence*, volume 2821 of *Lecture Notes in Computer Science*, pages 19–33. Springer Berlin Heidelberg, 2003. 16

[44] Thomas Hofmann. Probabilistic latent semantic indexing. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '99, pages 50–57, New York, NY, USA, 1999. ACM. 73

[45] Vahid Jalali and Mohammad Reza Matash Borujerdi. Information retrieval with concept-based pseudo-relevance feedback in medline. *Knowl. Inf. Syst.*, 29(1):237–248, October 2011. 74

[46] Frederick Jelinek and Robert L. Mercer. Interpolated estimation of Markov source parameters from sparse data. In *In Proceedings of the Workshop on Pattern Recognition in Practice*, pages 381–397, 1980. 53

[47] Antonio Jimeno-Yepes, Ernesto Jiménez-Ruiz, Rafael Berlanga Llavori, and Dietrich Rebholz-Schuhmann. Reuse of terminological resources for efficient ontological engineering in Life Sciences. *BMC Bioinformatics*, 10(S-10):4, 2009. 45

[48] Antonio Jimeno-Yepes, María Pérez, and Rafael Berlanga. Disambiguating automatically-generated semantic annotations for Life Science open registries. In *2nd International Workshop on Exploiting Large Knowledge Repositories (E-LKR'12)*, 2012. 124, 127

[49] Nick Juty, Nicolas Le Novre, and Camille Laibe. Identifiers.org and MIRIAM Registry: community resources to provide persistent identification. *Nucleic Acids Research*, 40(D1):D580–D586, 2012. 19, 22

[50] José Kahan and Marja-Ritta Koivunen. Annotea: an open RDF infrastructure for shared Web annotations. In *Proceedings of the 10th International Conference*

on World Wide Web, WWW '01, pages 623–632, New York, NY, USA, 2001. ACM. 16

[51] Mat Kala, Plebani Puntervoll, Alexandre Joseph, Edita Bartaeviit, Armin Tpfer, Prabakar Venkataraman, Steve Pettifer, Jan Christian Bryne, Jon Ison, Christophe Blanchet, Kristoffer Rapacki, and Inge Jonassen. BioXSD: the common data-exchange format for everyday bioinformatics web services. *Bioinformatics*, 26:540–546, 2010. 23

[52] Maryam Karimzadehgan and ChengXiang Zhai. Axiomatic analysis of translation language model for information retrieval. In *Proceedings of the 34th European conference on Advances in Information Retrieval*, ECIR'12, pages 268–280, Berlin, Heidelberg, 2012. Springer-Verlag. 60

[53] Vitaly Klyuev and Maxim Mozgovoy. Advances in Semantic Information Retrieval. *An International Journal of Computing and Informatics*, 35(4):399–533, 2011. 2

[54] Craig Knox, Savita Shrivastava, Paul Stothard, Roman Eisner, and David S. Wishart. BioSpider: A Web Server for Automating Metabolome Annotations. In *Pacific Symposium on Biocomputing*, pages 145–156. World Scientific, 2007. 19

[55] Shahad Kudama, Rafael Berlanga Llavori, Lisette Garcia-Moya, Victoria Nebot, and María José Aramburu. Towards Tailored Semantic Annotation Systems from Wikipedia. In *22nd International Workshop on Database and Expert Systems Applications (DEXA), 2011*, pages 478 –482, 2011. 44

[56] John Lafferty and Chengxiang Zhai. Document language models, query models, and risk minimization for information retrieval. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '01, pages 111–119. ACM, 2001. 54, 72

[57] Thomas K. Landauer, Peter W. Foltz, and Darrell Laham. An Introduction to Latent Semantic Analysis. *Discourse Processes*, (25):259–284, 1998. 73

[58] Victor Lavrenko and W. Bruce Croft. Relevance based language models. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '01, pages 120–127, New York, NY, USA, 2001. ACM. 57

[59] Ignacio Marín. Ancient origin of the Parkinson disease gene LRRK2. *Journal of Molecular Evolution*, 64:41–50, 2008. 113

[60] Victoria Martín-Requena, Javier Ríos, Maximiliano García, Sergio Ramírez, and Oswaldo Trelles. jORCA: easily integrating bioinformatics Web Services. *BioInformatics*, 26(4):553–559, 2010. 24

[61] Seeger Matthias. Learning with labeled and unlabeled data. Technical report, University of Edinburgh, 2000. 54

[62] Marco Mesiti, Ernesto Jiménez-Ruiz, Ismael Sanz, Rafael Berlanga, Giorgio Valentini, Paolo Perlasca, and David Manset. Data Integration Issues and Opportunities in Biological XML Data Management. In *Open and Novel Issues in XML Database Applications: Future Directions and Advanced Technologies*. IGI Global, 2009. 19

[63] David R. H. Miller, Tim Leek, and Richard M. Schwartz. A hidden Markov model information retrieval system. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '99, pages 214–221, New York, NY, USA, 1999. ACM. 53

[64] David Mimno and Andrew McCallum. Organizing the OCA: learning faceted subjects from a library of digital books. In *JCDL '07: Proceedings of the 7th ACM/IEEE-CS joint conference on Digital libraries*, pages 376–385, New York, NY, USA, 2007. ACM. 73, 74

[65] David Mimno, Hanna M. Wallach, Edmund Talley, Miriam Leenders, and Andrew McCallum. Optimizing semantic coherence in topic models. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, pages 262–272. Association for Computational Linguistics, 2011. 74

[66] Anaïs Mottaz, Yum Lina Yip, Patrick Ruch, and Anne-Lise Veuthey. Mapping proteins to disease terminologies: from UniProt to MeSH. *BMC Bioinformatics*, 9(S-5), 2008. 47

[67] Ismael Navas-Delgado. *An Infrastructure for Developing Applications in the Semantic Web*. PhD thesis, Universidad de Mlaga, 2009. 20

[68] Ismael Navas-Delgado, Maria del Mar Rojano-Muñoz, Sergio Ramírez, Antonio J. Pérez, Eduardo Andrés León, Jose F. Aldana-Montes, and Oswaldo Trelles. Intelligent client for integrating bioinformatics services. *Bioinformatics*, 22(1):106–111, 2006. 24

[69] David Newman, Jey Han Lau, Karl Grieser, and Timothy Baldwin. Automatic evaluation of topic coherence. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, HLT '10, pages 100–108, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics. 74

[70] John Niekrasz and Alexander Gruenstein. NOMOS: A Semantic Web Software Framework for Annotation of Multimodal Corpora. In *5th International Conference on Language Resources and Evaluation. LREC*, 2006. 16

[71] J. R. Norris. *Markov Chains (Cambridge Series in Statistical and Probabilistic Mathematics)*. Cambridge University Press, July 1998. 62

[72] Natalya F. Noy, Nigam H. Shah, Patricia L. Whetzel, Benjamin Dai, Michael Dorf, Nicholas Griffith, Clement Jonquet, Daniel L. Rubin, Margaret-Anne Storey, Christopher G. Chute, and Mark A. Musen. BioPortal: ontologies and integrated data resources at the click of a mouse. *Nucleic Acids Research*, 2009. 19

[73] Thomas Oinn, Mark Greenwood, Matthew Addis, Nedim Alpdemir, Justin Ferris, Kevin Glover, Carole Goble, Antoon Goderis, Duncan Hull, Darren Marvin, Peter Li, Phillip Lord, Matthew Pocock, Martin Senger, Robert Stevens, Anil Wipat, and Christopher Wroe. Taverna: lessons in creating a workflow environment for the life sciences. *Concurrency and Computation: Practice and Experience*, 18(10):1067–1100, August 2006. 26

[74] Artimo Panu, Jonnalagedda Manohar, Arnold Konstantin, Baratin Delphine, Csardi Gabor, de Castro Edouard, Duvaud Séverine, Flegel Volker, Fortier Arnaud, Gasteiger Elisabeth, Grosdidier Aurélien, Hernandez Céline, Ioannidis Vassilios, Kuznetsov Dmitry, Liechti Robin, Moretti Sébastien, Mostaguir Khaled, Redaschi Nicole, Rossier Grégoire, Xenarios Ioannis, and Stockinger Heinz. ExPASy: SIB bioinformatics resource portal. *Nucleic Acids Research*, 40:597–603, 2012. 19, 28

[75] Deuk Hee Park, Hyea Kyeong Kim, Il Young Choi, and Jae Kyeong Kim. A literature review and classification of recommender systems research. *Expert Systems with Applications*, 39(11):10059 – 10072, 2012. 10

[76] María Pérez, Rafael Berlanga, and Ismael Sanz. A semantic approach for the requirement-driven discovery of web services in the Life Sciences. In *3rd International Workshop on Semantic Web Applications and Tools for Life Sciences (SWAT4LS'10)*, 2010. 126

[77] María Pérez, Rafael Berlanga, Ismael Sanz, and María José Aramburu. Exploiting text-rich descriptions for faceted discovery of web resources. In *4th International Workshop on Semantic Web Applications and Life Sciences (SWAT4LS'11)*, 2011. 126, 127

[78] María Pérez, Rafael Berlanga, Ismael Sanz, and María José Aramburu. A semantic approach for the requirement-driven discovery of web resources in the Life Science. *Knowledge and Information Systems.*, 34(3):671–690, 2013. 127

[79] María Pérez, Rafael Berlanga, Ismael Sanz, and María José Aramburu. BioUSeR: A semantic-based tool for retrieving Life Science resources driven by text-rich user requirements. *Journal of Biomedical Semantics*, 4(12), May 2013. 127

[80] María Pérez, Sven Casteleyn, Ismael Sanz, and María José Aramburu. Requirements gathering in a model-based approach for the design of multi-similarity systems. In *Proceedings of the first international workshop on Model driven service engineering and data quality and security*, MoSE+DQS '09, pages 45–52, New York, NY, USA, 2009. ACM. 126

[81] María Pérez, Lisette García-Moya, and Rafael Berlanga. A translation model for facet-based retrieval in open registries. In *II Congreso Español de Recuperación de Información*, 2012. 126, 127

[82] María Pérez, Ismael Sanz, and Rafael Berlanga. A biclustering-based technique for requirement-driven Web Service selection. In *Actas de las JISBD 2010*, pages 109–120, 2010. 128

[83] María Pérez, Ismael Sanz, Rafael Berlanga, and María José Aramburu. Adding Semantics to the Discovery of Web Services. In *9th International Conference on Practical Applications of Agents and Multi-Agents Systems*, 2011. 126

[84] María Pérez, Ismael Sanz, Rafael Berlanga Llavori, and María José Aramburu. Semi-automatic Discovery of Web Services Driven by User Requirements. In *DEXA 2010*, pages 62–75, 2010. 127

[85] Steve Pettifer, Jon Ison, Matus Kalas, Dave Thorne, Philip McDermott, Inge Jonassen, Ali Liaquat, José M. Fernández, Jose M. Rodriguez, INB Partners, David G. Pisano, Christophe Blanchet, Mahmut Uludag, Peter Rice, Edita Bartaseviciute, Kristoffer Rapacki, Maarten Hekkelman, Olivier Sand, Heinz Stockinger, Andrew B. Clegg, Erik Bongcam-Rudloff, Jean Salzemann, Vincent Breton, Teresa K. Attwood, Graham Cameron, and Gert Vriend. The EMBRACE web service collection. *Nucleic Acids Research*, 38(suppl 2):W683–W688, 2010. 23

[86] Steve Pettifer, David Thorne, Philip McDermott, Terri Attwood, J. Baran, Jan Christian Bryne, Taavi Hupponen, D. Mowbray, and Gert Vriend. An Active Registry for Bioinformatics Web Services. *Journal Bioinformatics*, 25(16):2090–2091, 2009. 23, 44

[87] Natalia Ponomareva, Ferran Pla, Antonio Molina, and Paolo Rosso. Biomedical named entity recognition: a poor knowledge hmm-based approach. In *Proceedings of the 12th international conference on Applications of Natural Language to Information Systems*, NLDB'07, pages 382–387, Berlin, Heidelberg, 2007. Springer-Verlag. 46

[88] Andreas Prlic, Thomas Down, Eugene Kulesha, Robert Finn, Andreas Kahari, and Tim Hubbard. Integrating sequence and structural biology with DAS. *BMC Bioinformatics*, 8(1):333, 2007. 19, 21

[89] Sergio Ramírez, Antonio Muoz-Mérida, Johan Karlsson, Maximiliano García, Antonio J. Pérez-Pulido, M. Gonzalo Claros, and Oswaldo Trelles. MOWServ: a web client for integration of bioinformatic resources. *Nucleic Acids Research*, 38:W671–W676, 2010. 24

[90] Shuping Ran. A model for web services discovery with QoS. *SIGecom Exch.*, 4(1):1–10, March 2003. 17

[91] Dietrich Rebholz-Schuhmann, Miguel Arregui, Sylvain Gaudan, Harald Kirsch,

and Antonio Jimeno. Text processing through Web services: calling Whatizit. *Bioinformatics*, 24(2):296–298, 2008. 19

[92] Dietrich Rebholz-Schuhmann, Antonio José Jimeno Yepes, Erik M. Van Mulligen, Ning Kang, Jan Kors, David Milward, Peter Corbett, Ekaterina Buyko, Elena Beisswanger, and Udo Hahn. CALBC Silver Standard Corpus. *Journal of Bioinformatics and Computational Biology*, 08(01):163–179, 2010. 46

[93] Patrick Ruch. Automatic assignment of biomedical categories: toward a generic approach. *Bioinformatics*, 22(6):658–664, 2006. 19

[94] Giovanni Maria Sacco and Yannis Tzitzikas. *Dynamic Taxonomies and Faceted Search: Theory, Practice, and Experience*, volume 25. Springer, 2009. 56

[95] Gerard Salton. *The SMART Retrieval System - Experiments in Automatic Document Processing*. Prentice-Hall, Inc., 1971. 70

[96] Gerard Salton and Michael J. McGill. *Introduction to Modern Information Retrieval*. McGraw-Hill, Inc., New York, NY, USA, 1986. 14, 71

[97] Gerard Salton, Andrew Wong, and Chung Shu Yang. A vector space model for automatic indexing. *Communications of the ACM*, 18(11):613–620, 1975. 14

[98] Fabrizio Sebastiani. Machine learning in automated text categorization. *ACM Comput. Surv.*, 34(1):1–47, March 2002. 54

[99] Frank Spitzer. *Principles of random walk*. Springer-Verlag, 1976. 55

[100] Keith Stevens, Philip Kegelmeyer, David Andrzejewski, and David Buttler. Exploring Topic Coherence over many models and many topics. In *The 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 952–961. 2012 Association for Computational Linguistics, 2012. 74

[101] Mark Steyvers and Tom Griffiths. *Probabilistic Topic Models*. Lawrence Erlbaum Associates, 2007. 54, 72

[102] Chenliang Sun, Yi Lin, and Bettina Kemme. Comparison of UDDI Registry Replication Strategies. In *Proceedings of the IEEE International Conference on Web Services*, ICWS '04, pages 218–. IEEE Computer Society, 2004. 12

[103] Dat Tran, Christopher Dubay, Paul Gorman, and William Hersh. Applying Task Analysis to Describe and Facilitate Bioinformatics Tasks. *MEDINFO*, 107(Pt 2):818, 2004. 54, 88

[104] Benjamin P. Vandervalk, E. Luke McCarthy, and Mark D. Wilkinson. Moby and Moby 2: Creatures of the Deep (Web). *Briefings in Bioinformatics*, pages 114–128, 2009. 23

[105] Hanna M. Wallach, Iain Murray, Ruslan Salakhutdinov, and David Mimno. Evaluation methods for topic models. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, pages 1105–1112, New York, NY, USA, 2009. ACM. 74

[106] Yiqiao Wang and E. Stroulia. Flexible interface matching for Web-service discovery. In *Web Information Systems Engineering, 2003. WISE 2003. Proceedings of the Fourth International Conference on*, pages 147 – 156, dec. 2003. 15

[107] Xing Wei and W. Bruce Croft. LDA-based document models for ad-hoc retrieval. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '06, pages 178–185, 2006. 73

[108] Mark D. Wilkinson and Matthew Links. BioMOBY: An open source biological web services proposal. *Briefings in Bioinformatics*, 3(4):331–341, 2002. 2, 19, 23

[109] Mark D. Wilkinson, Benjamin Vandervalk, and Luke McCarthy. The Semantic Automated Discovery and Integration (SADI) Web service Design-Pattern, API and Reference Implementation. *Journal of Biomedical Semantics*, 2:8, 2011. 19, 20, 24

[110] Bian WU and Xincai WU. A QoS-aware Method for Web Services Discovery. *Journal of Geographic Information Systems*, 2:40–44, 2010. 18

[111] Lei Ye and Bin Zhang. Discovering Web Services Based on Functional Semantics. In *Proceedings of the 2006 IEEE Asia-Pacific Conference on Services Computing*, APSCC '06, pages 348–355, Washington, DC, USA, 2006. IEEE Computer Society. 17

[112] Antonio Jimeno Yepes. *Ontology Refinement for Improved Information Retrieval in the Biomedical Domain*. PhD thesis, Universitat Jaume I, 2009. 74

[113] Daisuke Yokomoto, Kensaku Makita, Hiroko Suzuki, Daichi Koike, Takehito Utsuro, Yasuhide Kawada, and Tomohiro Fukuhara. Lda-based topic modeling in labeling blog posts with wikipedia entries. In Hua Wang, Lei Zou, Guangyan Huang, Jing He, Chaoyi Pang, HaoLan Zhang, Dongyan Zhao, and Zhuang Yi, editors, *Web Technologies and Applications*, volume 7234 of *Lecture Notes in Computer Science*, pages 114–124. Springer Berlin Heidelberg, 2012. 73

[114] Daisuke Yokomoto, Kensaku Makita, Hiroko Suzuki, Daichi Koike, Takehito Utsuro, Yasuhide Kawada, and Tomohiro Fukuhara. LDA-Based Topic Modeling in Labeling Blog Posts with Wikipedia Entries. In Hua Wang, Lei Zou, Guangyan Huang, Jing He, Chaoyi Pang, HaoLan Zhang, Dongyan Zhao, and Zhuang Yi, editors, *Web Technologies and Applications*, volume 7234 of *Lecture Notes in Computer Science*, pages 114–124. Springer Berlin Heidelberg, 2012. 73, 74

[115] Eric Yu. *Modelling Strategic Relationships for Process Reenginering*. PhD thesis, University of Toronto, Canada, 1995. 35

[116] Eric Yu. Towards Modelling and Reasoning Support for Early-Phase Requirements Engineering. In *RE 1997*, volume 85, pages 2444–2448, 1997. 35

[117] Jie Zhang, Dan Shen, Guodong Zhou, Jian Su, and Chew-Lim Tan. Enhancing hmm-based biomedical named entity recognition by studying special phenomena. *J. of Biomedical Informatics*, 37(6):411–422, December 2004. 46

[118] Shibin Zhou, Kan Li, and Yushu Liu. Text categorization based on topic model. In *Proceedings of the 3rd international conference on Rough sets and knowledge technology*, RSKT'08, pages 572–579, Berlin, Heidelberg, 2008. Springer-Verlag. 73