UNIVERSIDADE DE LISBOA
FACULDADE DE CIÊNCIAS
DEPARTAMENTO DE INFORMÁTICA

Automated Extension of Biomedical Ontologies

Cátia Luísa Santana Calisto Pesquita

DOUTORAMENTO EM INFORMÁTICA
ESPECIALIDADE BIOINFORMÁTICA

2012

UNIVERSIDADE DE LISBOA
FACULDADE DE CIÊNCIAS
DEPARTAMENTO DE INFORMÁTICA

Automated Extension of Biomedical Ontologies

Cátia Luísa Santana Calisto Pesquita

DOUTORAMENTO EM INFORMÁTICA
ESPECIALIDADE BIOINFORMÁTICA

Tese orientada pelo Dr. Francisco José Moreira Couto

2012

Abstract

Developing and extending a biomedical ontology is a very demanding process, particularly because biomedical knowledge is diverse, complex and continuously changing and growing. Existing automated and semi-automated techniques are not tailored to handling the issues in extending biomedical ontologies.

This thesis advances the state of the art in semi-automated ontology extension by presenting a framework as well as methods and methodologies for automating ontology extension specifically designed to address the features of biomedical ontologies.The overall strategy is based on first predicting the areas of the ontology that are in need of extension and then applying ontology learning and ontology matching techniques to extend them. A novel machine learning approach for predicting these areas based on features of past ontology versions was developed and successfully applied to the Gene Ontology. Methods and techniques were also specifically designed for matching biomedical ontologies and retrieving relevant biomedical concepts from text, which were shown to be successful in several applications.

Keywords: ontology extension, ontology matching, ontology learning, ontology evolution

Resumo

O desenvolvimento e extensão de uma ontologia biomédica é um processo muito exigente, dada a diversidade, complexidade e crescimento contínuo do conhecimento biomédico. As técnicas existentes nesta área não estão preparadas para lidar com os desafios da extensão de uma ontologia biomédica.
Esta tese avança o estado da arte na extensão semi-automática de ontologias, apresentando uma *framework* assim como métodos e metodologias para a automação da extensão de ontologias especificamente desenhados tendo em conta as características das ontologias biomédicas. A estratégia global é baseada em primeiro prever quais as áreas da ontologia que necessitam extensão, e depois usá-las como enfoque para técnicas de alinhamento e aprendizagem de ontologias, com o objectivo de as estender. Uma nova estratégia de aprendizagem automática para prever estas áreas baseada em atributos de antigas versões de ontologias foi desenvolvida e testada com sucesso na Gene Ontology. Foram também especificamente desenvolvidos métodos e técnicas para o alinhamento de ontologias biomédicas e extracção de conceitos relevantes de texto, cujo sucesso foi demonstrado em várias aplicações.

Palavras Chave: extenso de ontologias, alinhamento de ontologias, aprendizagem de ontologias, evoluo de ontologias

Resumo Estendido

As ontologias biomédicas representam um importante avanço em Bioinformática, uma vez que auxiliam num dos grandes desafios desta área, a gestão e extracção de conhecimento. O desenvolvimento de uma ontologia biomédica é um processo muito exigente, dada a diversidade, complexidade e crescimento contínuo do conhecimento biomédico. Um dos maiores desafios na manutenção de uma ontologia é mantê-la actualizada. Com o advento de técnicas como a sequenciação automática de DNA e *microarrays* tem sido gerada uma grande quantidade de dados que resultaram num crescimento exponencial do número de publicações científicas e dos dados armazenadas em bases de dados biomédicas. As ontologias biomédicas, em especial as mais bem-sucedidas, são utilizadas todos os dias por investigadores de várias áreas, e necessitam estar actualizadas para cumprirem o seu propósito.

O desenvolvimento de uma ontologia continua a ser um esforço essencialmente manual, especialmente no caso das ontologias biomédicas dado o seu domínio complexo. No entanto, têm sido desenvolvidas na última década várias técnicas automáticas ou semi-automáticas para auxiliar na construção e manutenção de uma ontologia. Estas técnicas têm sido essencialmente aplicadas em ontologias dedicadas a domínios muito mais restritos e simples que os das ontolgias biomédicas. Muitas destas técnicas utilizam propriedades de ontologias formais, enquanto que as ontologias biomédicas têm normalmente uma estrutura simples. Adicionalmente, muitas usam também processamento e análise de texto, o que é um desafio especial no domínio biomédico, dada a complexidade e ambiguidade da terminologia biomédica. Não obstante, o grande volume de literatura biomédica disponível e a proliferação de ontologias biomédicas e outros recursos do género, encorajam

o desenvolvimento e adaptação de técnicas de extensão de ontologias, capazes de encontrar novos conceitos para representar novo conhecimento.

Esta tese debruça-se sobre o tema da extensão de ontologias no contexto das ontologias biomédicas. O seu objectivo é o desenvolvimento de uma *framework* para a extensão semi-automática de ontologias assim como métodos e metodologias que auxiliem na automação de alguns dos processos de extensão de modo a aliviar o esforço dos curadores. A *framework* proposta centra-se nos desafios específicos da extensão de ontologias biomédicas, oferecendo três componentes: previsão de extensão, aprendizagem e alinhamento. A previsão de extensão ataca os problemas relacionados com a grande quantidade de literatura e ontologias relacionadas disponível, ao identificar áreas que necessitam de extensão. Estas são usadas para focar os esforços de aprendizagem e alinhamento que geram listas de conceitos candidados ao explorar a abundante literatura e ontologias.
Dada a existência de vários sistemas para aprendizagem e alinhamento de ontologias, o enfoque principal da tese é no desenvolvimento de métodos para a componente de prever extensão. Eu desenvolvi e testei duas estratégias: uma de regras de outra de aprendizagem supervisionada. A estratégia de regras é baseada em orientações para o desenvolvimento de ontologias e provou não ser apropriada para a tarefa. A estratégia de aprendizagem supervisionada é baseada na noção de que é possível aprender um modelo que distingue entre as áreas que irão ser estendidas e aquelas que não o serão, baseado em características das ontologias. Esta estratégia atingiu os 79% de *f-measure* na previsão do refinamento de uma porção do ramo *biological process* da Gene Ontology (GO).
Desenvolvi também métodos e técnicas para aprendizagem e alinhamento de ontologias de modo a garantir o seu sucesso.
No que diz respeito à aprendizagem, desenvolvi uma medida de relevância de termos capaz de ordenar conceitos candidatos de acordo com

a sua relevância para o domínio. Isto é particularmente relevante em domínios altamente específicos e complexos como a biomedicina. A medida prosposta, FLOR, usa o conteúdo de evidência das palavras no vocabulário da ontologia para medir a relação entre conceitos da ontologia e candidatos. O FLOR foi também aplicado para medir a relação entre conceitos da ontologia de modo a gerar novas relações.

Dois conjuntos de métodos foram desenvolvidos para alinhar ontologias, sendo posteriormente submetidos ao OAEI (Iniciativa para Avaliação de Alinhamento de Ontologias). O primeiro, baseado em semelhança léxica e técnicas de computação de semelhança global, obteve bons resultados mas não superou o estado da arte. O segundo, resultante de uma colaboração com a equipa do AgreementMaker, obteve o primeiro lugar na competição com uma *f-measure* de 91.7% no alinhamento de ontologias da anatomia. Os métodos empregues resultaram de uma série de melhorias aos métodos de semelhança léxica que exploraram os desafios e cracterísticas das ontologias biomédicas. Um destes métodos é baseado na adição de novos sinónimos à ontologia e é portanto também um método para enriquecimento da ontologia. Com o inuito de testar a aplicação destes métodos noutro domínio, foi também realizado o alinhamento de uma porção do GO com o FMA (Foundational Model of Anatomy), onde foi obtida uma *f-measure* de 90.7%.

As contribuições desta tese não são apenas originais na sua essência mas possuem também um objectivo complexo, representando portanto aproximações válidas à resolução dos desafios da extensão de ontologias biomédicas. A interação dos métodos aqui descritos com sistemas já existentes de acordo com a estrutura da *framework* proposta, resulta numa metodologia para a extensão de ontologias capaz de ser aplicada a ontologias biomédicas ou outras ontologias com características semelhantes.

A extensão de ontologias é uma tarefa esencial da engenharia de ontologias e a automatização de alguns dos seus processos pode contribuir não só para a diminuição do investimento de recursos mas também para garantir uma actualização atempada da ontologia, o que pode ser crucial em áreas de desenvolvimento rápido como a genómica, a epidemiologia ou a saúde. Considero que o futuro do desenvolvimento de ontologias passa necessariamente por uma automatização de alguns dos seus processos, nomeadamente os mais entediantes e demorados, libertando assim os peritos em ontologias para se focarem em aspectos de modelação mais complexos. É a integração bem sucedida do conhecimento humano com métodos automáticos que irá garantir a realização do potencial das ontologias biomédicas como ferramentas essenciais para lidar com os desafios de gestão de conhecimento nas Ciências da Vida no século XXI, e para a qual considero que esta tese é um avanço.

Acknowledgements

This thesis would not have been possible without the help and encouragement of many people.

First, I would like to express my sincere gratitude to my advisor Prof. Francisco Couto, for his constant guidance and support, good-humored encouragement and for believing in me. His patience and knowledge were invaluable and one simply could not wish for a better or friendlier supervisor.

I would also like to thank Prof. Mário Silva who was my co-supervisor for the first part of my thesis and who gave me priceless suggestions and advice. It was a privilege to have worked with him.

I am grateful to the Fundação para a Ciência e Tecnologia, for funding my PhD scholarship SFRH/BD/42481/2007, without which my research would not have been possible.

I am also grateful to Prof. André Falcão for his lively discussions and to Prof. António Branco for teaching me how to write code, and believing a Biologist could do it.

I am very grateful to Daniel Faria, whose clever discussions contributed both to this thesis and to many other interesting projects.
I am also grateful to Tiago Grego who contributed directly to this thesis by using one of its methods in his work and to Hugo Bastos,

Bruno Tavares and João Ferreira with whom I collaborated in many projects and fun breaks, and also to Cátia Machado for all the discussions on both research and non-research topics, which made it easier to get through the day. And of course I could not forget everyone else at LaSIGE who contributed to a great working environment.

I am particularly grateful to Prof. Isabel Cruz for welcoming me to the ADVIS lab at the University of Chicago and being an inspiration for women in Computer Science and to Cosmin Stroe at the ADVIS lab, for his incredible drive, quick thinking and good humor, without which we would never have won OAEI.

A special thanks goes to all my friends, in particular Ivo and Susana for helping me with my son while I was writing this thesis, and for helping me relax when I took breaks.

I am infinitely grateful to my family, especially my parents who encouraged me to pursuit science in the first place and whose love, support and encouragement never faltered, and my sister for sharing everything with me since we were little.
I would not have made it without my husband José who stood by me every step of the way, made me laugh at myself and believed in me when I needed it the most.
Finally, I am lovingly grateful to my son Artur, for all his smiles and all the naps he took while I was writing this thesis.

June 16, 2012

Catia Pesquita

To José and Artur

Contents

List of Figures

LIST OF FIGURES

List of Tables

Chapter 1

Introduction

In recent years biomedical research has been generating an enormous amount of data due to the advent of high throughput techniques such as automated DNA sequencing and microarrays. This data deluge brought on the emergence of Bioinformatics, as computers became essential to store, manage and analyze the ever-increasing amount of data. In spite of last decades efforts to structure and organize biomedical data, there are still many issues that challenge biomedical knowledge discovery and management (Rubin *et al.*, 2008). On one hand, most scientific knowledge is still present only in natural language text in the form of scientific publications, whose number grows exponentially. The alarming growth rate in the number of publications makes it necessary to employ text mining techniques if we are ever to aspire at keeping up with their speed. However, the natural ambiguity and subjectivity of natural language hinders the automated processing of scientific publications. Although there have been some well-intended discussions on the enforcing of structured digital abstracts by publishers (Gerstein *et al.*, 2007), this is still only a conjecture. On the other hand, although there is a large number of databases to store biomedical data, the effort to achieve interoperabiltity between them is still lagging behind, given that most resources, particularly the older ones, were developed in a completely independent fashion, and only in the last few years has there been an effort to connect them to other resources. One very important breakthrough for both areas, was the development of biomedical ontologies (bio-ontologies). They support both issues by providing unequivocal and structured models of specific domains, which

is fundamental to resolve semantic ambiguities in text mining and also to serve as a common background to biomedical databases.
However, the development of a bio-ontology, or other domain ontologies, is a very demanding process that requires both expertise in the domain to model, as well as in ontology design. This means that people from very different backgrounds, such as biology, philosophy and computer science should be involved in the process of creating an ontology. However, many specific bio-ontologies are built by small teams of life sciences researchers, with little experience in ontology design. Ontology developers are responsible for first, agreeing on the precise limits of the domain to model; second, defining the structure and complexity of the model; and finally, building the ontology itself by creating the concepts, relations and other axioms it might contain (Aranguren *et al.*, 2008). Several methodologies have been developed to help build ontologies (Aranguren *et al.*, 2008; Denicola *et al.*, 2009; Pinto *et al.*, 2004; Sure *et al.*, 2003). Nevertheless, ontology development remains a manual and labor intensive task, with ontology engineers traditionally not concerned with the effort involved in it, given that once an ontology is built, the task is finished. This is not the case in the Life sciences domain, where knowledge is diverse, complex and continuously changing and growing. Bio-ontologies can never be considered complete, but always having to adapt to our new understanding of biological knowledge. This forces bio-ontology development to be an iterative process (Pinto *et al.*, 2009), to keep up with the dynamic and evolving domain. This process, usually named ontology evolution, is a continuous effort, requiring large investments of both time and money with each new version that is produced. Moreover, many bio-ontologies cover large and complex domains which magnifies the effort required, even when considering highly successful ontologies, such as the Gene Ontology (GO Consortium, 2010), where a large community is engaged in its creation. These challenges create the need for semi-automated systems that are able to support ontology engineers in the task of ontology evolution, and in particular in ontology extension, the most frequent and time consuming evolution task in biomedical ontologies.

The aim of this thesis is the development of methods and methodologies to be integrated in a proposed framework for semi-automated ontology extension

that aims at alleviating the burden on biomedical ontology developers. One of the main challenges these experts face is the size and complexity of the textual corpora they need to analyze in order to create new ontology versions. These methods are specifically tailored to handle the characteristics of bio-ontologies and the life sciences domain, by leveraging on the large amount of publicly available biomedical literature and the many biomedical ontologies and terminologies, through text mining, machine learning, ontology learning and ontology matching techniques. The main testing ground for the developed methods is the Gene Ontology (GO), which is currently the most successful case of ontology application in molecular biology and provides an ontology for functional annotation of gene-products in a cellular context, capable of dealing with the semantic heterogeneity of gene product annotations.

1.1 Research Questions and Contributions

In the following, an overview of this work is presented in terms of the research questions that were addressed and the accomplished contributions.

1.1.1 Research Questions

The research questions addressed in this thesis span three areas of ontology engineering: ontology extension, ontology matching and ontology learning. Ontology matching and learning provide methods that can be used in the field of ontology extension.

Biomedical Ontology Extension One of the research topics in this thesis is what are the specific challenges of extending biomedical ontologies (I). Researching these issues is a cornerstone for the development of a framework for the extension of biomedical ontologies (II). A relevant issue in biomedical ontology extension is the identification of the needed changes, which is rendered more difficult by the size and complexity of the domain. So another question this thesis addresses is if it is possible to automate the change capturing phase of ontology extension and thus identify the ontology areas that need to be extended(III).

Ontology Learning A crucial challenge in extracting biomedical ontology concepts from text is the differentiation between general terms and domain terms. In this thesis I address the question of wether it is possible to support this differentiation by exploring the vocabulary of the ontology to extend (IV).

Ontology Matching In this area I investigate what are the specific issues in matching biomedical ontologies (V) and if current ontology matching methods can be improved to handle them (VI). And finally if this improvement can be achieved without using external resources (VII).

1.1.2 Contributions

When choosing a research topic for my doctoral thesis I was motivated by my previous work in semantic similarity for my MSc dissertation. An initial exploration of semantic similarity measures resulted in the publication of a review on PLoS Computational Biology (Pesquita *et al.*, 2009b), the highest impact factor journal in Bioinformatics. Semantic similarity was subsequently applied in two areas of my work: ontology matching and candidate term filtering. This thesis advances the state of the art in semi-automated biomedical ontology extension by presenting a framework as well as methods and methodologies for ontology extension specifically designed to address the issues of extending biomedical ontologies:

Framework for automating ontology extension (II) which integrates three main components: extension prediction, learning and matching. I designed this framework to solve some of the issues in extending biomedical ontologies and also to support the integration of novel and existing automated methods with human expert verification.

Conceptual framework for analysis of ontology extension (I) , where I propose a series of guidelines for analyzing ontology extension. The application of these guidelines to the Gene Ontology revealed interesting patterns in its development (Pesquita & Couto, 2011).

Supervised learning approach for automating change capturing (III) , where I created a novel approach based on features of previous versions of the ontology to support the prediction of extension events in areas of the ontology. This approach reached 0.79 f-measure and also identified the minimal number of versions and features able to support prediction. The good results obtained with this approach contrasted with the poor results found by using rules based on traditional guidelines for ontology development. These results were published in PLoS Computational Biology (Pesquita & Couto, 2012) and in the International Conference on Biomedical Ontologies (Pesquita & Couto, 2011).

Novel methods for filtering candidate concepts extracted from text (IV) were proposed with the goal of ensuring the relevance of candidate concepts to the ontology. FLOR, the main method I developed, is a term relevance measure based on the ontology vocabulary which can be used to calculate the relatedness between ontology concepts. FLOR was applied to find relations between ontology concepts with the purpose of enriching the ontology and it was found to achieve a good performance. FLOR was also applied as an entity-resolution module in an approach for the recognition and resolution of chemical entities, where it was shown to outperform a dictionary-based approach by 2-5% f-measure (Grego *et al.*, 2012).

Novel and improved methods for ontology matching (V-VII) were developed with the specific intent of handling the issues I identified in the alignment of biomedical ontologies. These methods were submitted to OAEI 2010 and OAEI 2011. In OAEI 2010 my system reached 5th and 2nd place in two of the competition tasks (Pesquita *et al.*, 2010). In OAEI 2011, in collaboration with the AgreementMaker team, my contributions helped to achieve the 1st place in the competition, improving on past years results (Cruz *et al.*, 2011). Moreover, I also tested their application in aligning a portion of GO and FMA where an f-measure of 90.8% was achieved. It is also noteworthy that one of the proposed methods also functions as an ontology enrichment strategy by extending the ontology with new synonyms.

1.2 Reader's Guide

This thesis is organized in three main parts:

Foundation where an overview of the theoretical foundations needed for this thesis is given. Chapter 2 gives a short introduction to ontologies and ontology evolution, describing some biomedical ontologies as well. Chapter 3 focuses on relevant approaches in ontology extension, detailing the state of the art in ontology extension and its sister areas, learning and matching. Chapter 4 provides a basic introduction to machine learning and semantic similarity, two techniques employed in this thesis.

Methods The second part describes in detail the approaches developed to support the semi-automated extension of biomedical ontologies. In chapter 5 a framework for semi-automated extension is presented along with an analysis of challenges and opportunities for this area in the biomedical domain. It also describes a conceptual framework for the analysis of ontology extension and its application to the Gene Ontology. Chapter 6 presents the approaches developed for predicting ontology extension. Chapter 7 focuses on exploring ontology vocabulary in the context of ontology learning and enrichment while ontology matching is addressed in chapter 8.

Conclusions The third part concludes this thesis by presenting an overall summary, an outlook to future work and some final remarks.

Part I

Foundation

Chapter 2

Ontologies and Ontology Evolution

This chapter is divided in two main sections. The first is dedicated to the definition of ontology, and a description of bio-ontologies, with a focus on ontologies to be used throughout this work. The second describes the state of the art in ontology extension and provides a general delineation of the closely related areas of ontology learning and matching.

2.1 Ontologies

2.1.1 Background

The origins of ontology date back to Aristotle who first proposed the study of being and reality, and particularly the classification within a hierarchy of the entities that exist. In this sense, ontology is comparable in many ways to the taxonomies produced by the life sciences, dedicated to the description and classification of organisms, that first came into existence in the 18th century with Linnaeus.
But the concept of ontology as a technical term in the context of computer science was only coined in the early 1990's by Gruber (Gruber,

1993) as *"an explicit specification of a conceptualization of a domain"*. This definition caused much debate, so we refer to (Gruber, 2008) for a clear and more recent definition:

Definition 1. *"An ontology defines (specifies) the concepts, relationships, and other distinctions that are relevant for modeling a domain. The specification takes the form of the definitions of representational vocabulary (classes, relations, and so forth), which provide meanings for the vocabulary and formal constraints on its coherent use."*

This means that the modeling provided by an ontology should specify a systematic correlation between reality and its representation. It should also allow automatic information processing through its formalization, while remaining understandable and clear to a domain expert (Smith, 2003).

In Computer Science several kinds of data and conceptual models are considered to be ontologies, covering a wide range of expressiveness and level of axiomatization, including: glossaries, thesauri, XML schemas, database schemas, taxonomies and formal ontologies (axiomatised theories). The latter are typically encoded in formal languages that allow abstraction from lower level data, such as OWL, OBO, RDF Schema. Ontologies can function in multiple roles (Noy & Mcguinness, 2000b):

- To share common understanding of the structure of information among people or software agents;
- To enable reuse of domain knowledge;
- To make domain assumptions explicit;
- To separate domain knowledge from the operational knowledge;
- To analyze domain knowledge.

The core of all ontologies are their classes or concepts. These are usually organized in a hierarchical fashion, but can also be related through other types of relations. The instances of the classes can also be an important element of ontologies, and even if the ontology does

not model them, they are essential to materialize the concepts. A simple definition of these three ontology elements, concepts, relations and instances is given:

Definition 2. *An ontology class or concept provides the abstraction mechanism for grouping resources with similar characteristics. Classes have an intensional meaning (their underlying concept) which is related but not equal to their extension (the instances that compose the class).*

Definition 3. *An ontology relation is a binary relation established between classes or concepts.*

Definition 4. *An ontology instance or individual is an individual object pertaining to a domain.*

Ontologies are often depicted as labeled graphs where nodes represent the classes, and edges the relations between them.

The remainder of this section is dedicated to bio-ontologies, in particular the Gene Ontology, and to WordNet, a lexical ontology commonly used in ontological applications.

2.1.2 Biomedical Ontologies

One of the scientific areas where ontologies have had more success is biomedicine. The domain knowledge in this area is too vast to be dealt with by a single researcher. Therefore, there is a need to use approaches such as biomedical ontologies (ontologies applied to the biomedical domain), to handle the application of domain knowledge to biological data. The role of bio-ontologies has changed in recent years: from limited in scope and scarcely used by the community, to a main focus of interest and investment. Although clinical terminologies have been in use for several decades, different terminologies were used for several purposes, hampering the sharing of knowledge and its reliability. This has lead to the creation of bio-ontologies to answer the need to merge and organize the knowledge, and overcome the semantic heterogeneities observed in this domain. While the first attempts at

developing them focused on a global schema for resource integration, real success and acceptance was only achieved later by ontologies for annotating bioentities[1], namely the Gene Ontology (Bodenreider & Stevens, 2006). Since then, bio-ontologies have been used successfully for other goals, such as description of experimental protocols and medical procedures.

The maturity of biomedical ontologies is embodied in the goals of the OBO Foundry (Smith *et al.*, 2007), a self-appointed foundry responsible for the goal of creating a suite of orthogonal interoperable reference ontologies in the biomedical domain.

2.1.2.1 Gene Ontology

The Gene Ontology (GO) (GO Consortium, 2010) is a task-oriented ontology that was created for the functional annotation of gene products in a cellular context. This means that the concepts in the ontology are applied to describe aspects of gene product functions. A gene product can be defined as follows:

Definition 5. *A gene product is the result of the expression of a gene, either an RNA strand or a protein.*

Since GO is more commonly employed to annotate proteins, from now on we focus on these biomolecules. GO is only composed of classes, referred to as GO terms, and never the instances to which they apply, i.e. gene products.

GO is divided in three categories (or GO types), which constitute three ontologies:

- molecular function, which is dedicated to the description of processes at the molecular level;

[1]Biological entities, e.g. genes, proteins, anatomical parts, diseases

- biological process, which handles the assemblies of various molecular functions;
- cellular component, which is in charge of cellular locations and macromolecular complexes.

Each of these ontologies is composed of terms and relationships between them, which are structured as a Directed Acyclic Graph (DAG), where terms are nodes and relationships are edges (see Figure 2.1. A GO term is a natural language term (e.g. transport) with a corresponding unique seven-digit numeric identifier (e.g. GO:0006810) and a natural-language definition. The nearer a term is to the root of the graph, the more general it is (e.g. binding, intracellular) and traveling deeper into the graph, the terms become more specialized (e.g retinal binding, cytoplasmic nucleosome).

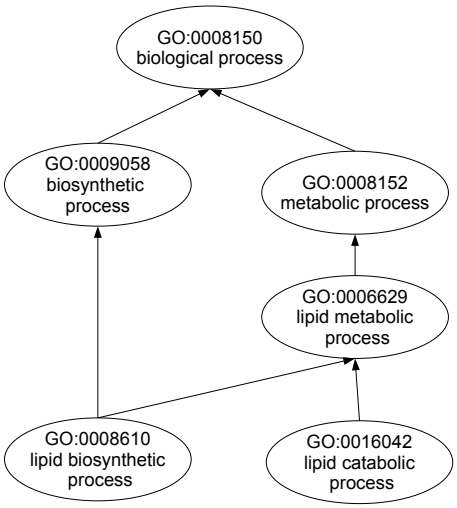

Figure 2.1: A subgraph of the biological process ontology from GO.

GO currently employs 6 distinct types of relations:

- *is_a* is a subsumption relationship: one term subsumes the other (e.g DNA binding is a subclass of binding, so DNA binding *is_a* binding). *is_a* is a transitive relationship, that is, a child is a subclass of its grandfather since its father is also a subclass of its grandfather.
- *part_of* expresses part-whole relationships, more explicitly necessarily part of, that is the existence of the whole does not imply the existence of the part, but wherever the part exists it exists as a part of the whole.
- *has_part* represents a part–whole relationship from the perspective of the parent, and is thus the logical complement to the *part_of* relationship.
- *regulates* is applied to situations where one process directly affects the manifestation of another process or quality. More specifically it means necessarily regulates: whenever B is present, it always regulates A, but A may not always be regulated by B.
- *positively_regulates* is a sub-relation of *regulates* that represents a more specific form of regulation.
- *negatively_regulates* is a sub-relation of *regulates* that represents a more specific form of regulation.

The relationship types *part_of* and *regulates* are also used to link terms from the molecular function and biological process ontologies.

Annotation of proteins with GO

The primary functionality of GO, the annotation of gene products, is largely achieved by the GOA project (Camon *et al.*, 2004), which provides GO term annotations for gene products present in UniProt, the largest repository of protein sequences, and other major databases. The functional annotation of proteins is the assignment of a GO term to a protein in order to describe an aspect of its function. In the context of GO, annotations are always accompanied by an evidence

code (a three letter acronym), which describes the kind of evidence that supported the annotation. GO evidence codes can be classified into two main categories: manual and electronic. Manual annotations correspond to those that were manually curated, whereas electronic annotations correspond to annotations automatically generated and comprise the vast majority of GO annotations. Furthermore, based on the *true path rule*, annotations to a GO term are automatically transmitted to its ancestors, so GO terms can be said to have two types of annotations: direct and inherited. Direct annotations correspond to annotations made directly to the term, whereas inherited annotations, are annotations made to their children. Finally, a protein may be annotated with as many GO terms as necessary to fully describe its functional aspects.

Development of GO

GO is a handcrafted ontology supported by OBO-Edit, an ontology editing software (Day-richter *et al.*, 2007). There are about 100 contributors to GO spread across the GO Consortium and several GO Associates members, and they are expected to contribute regularly towards the content of GO. Since GO covers a broad range of biological areas, GO has setup interest groups to discuss the areas within the ontology that are likely to require extensive additions or revisions. These groups roughly correspond to high-level terms: cardiovascular, developmental biology, electron transport, farm animals, immunology, metabolism, neurobiology, pathogens and pathogenesis, protein kinases, response to drug, and transport. Other GO users can also contribute by suggesting new terms via Sourceforge.net, however the majority of content requests are made by GO team members (Pesquita *et al.*, 2009a) (see Table **??**).

	people	total requests	request/person
GO members	53	2545	48.02
External users	46	337	7.33

Table 2.1: Summary of new GO term requests on Sourceforge.net[1] as of March 2009

2.1.2.2 Anatomy Ontologies

Anatomical representations are crucial to the integration of biomedical knowledge, since every biological process occurs in a given anatomical part. Several representation schemas have been created over the past fifteen years, ranging from simple lists of terms to complete ontologies. Three anatomical ontologies are particularly relevant for the context of this work: the Foundation Model of Anatomy, the Adult Mouse Anatomical Dictionary and the NCI Thesaurus Anatomical branch. The Foundation Model of Anatomy (FMA) (Rosse & Jr, 2003) is a domain ontology of the concepts and relationships that pertain to the structural organization of the human body. It was developed to function as a reference ontology in biomedical informatics for correlating different views of anatomy, aligning existing and emerging ontologies and providing a structure-based template for representing biological functions. The Adult Mouse Anatomical Dictionary (Hayamizu *et al.*, 2005) provides standardized nomenclature for anatomical terms in the postnatal mouse. It is structured as a directed acyclic graph, and is organized hierarchically both spatially e.g. *mitral valve is part of heart* and functionally e.g *heart is part of cardiovascular system*, using *is_a* and *part_of* relations. It contains more than 2,400 unique anatomical terms, 305 of which have also synonyms. The NCI Thesaurus (Sioutos *et al.*, 2007) provides a reference terminology that integrates molecular and clinical cancer-related information within a unified biomedical informatics framework. One of its branches is dedicated to Human anatomy and is composed of over 4,000 terms, of which over a quarter have synonyms.

Anatomical ontologies are particularly well suited to test ontology matching strategies, since there are direct correspondences between anatomical parts of distinct species. All three ontologies have been used in the OAEI (Ontology Alignment Evaluation Initiative) anatomy track (Ferrara *et al.*, 2009), but FMA is no longer a part of this challenge, though it has been used to support the matching of other anatomical ontologies (Zhang *et al.*, 2005).

2.1.3 WordNet

WordNet (Miller, 1995) is a lexical database for the English language designed to support automatic text analysis and artificial intelligence applications. It groups English words into sets of synonyms called synsets, and nouns and verbs are organized into hierarchies, defined by hypernym or is_a relationships.
Consequently, WordNet can be seen as a lexical ontology, where the hypernym/hyponym relationships between the nouns synsets can be interpreted as relations between ontology concepts. The WordNet ontology is commonly represented as a graph.
WordNet has had many different applications in information systems, including word sense disambiguation, information retrieval, text classification, text summarization, and ontology learning (Morato *et al.*, 2004).

2.2 Ontology Evolution

Ontology evolution can be defined as the process of modifying an ontology in response to a certain change in the domain or its conceptualization (Flouris *et al.*, 2008):

- changes in the domain, when new concepts belonging to the domain are added to reflect new knowledge or a re-purposing of the ontology

- changes in conceptualization, which can result from a changing view of the domain and from a change in usage perspective

In general, the evolution of bio-ontologies is mainly concerned with the first type, given the dynamic nature of biological knowledge production. Everyday new discoveries are published, rendering some facts obsolete and bringing new knowledge to light.

To tackle the complexity inherent to ontology evolution (Stojanovic *et al.*, 2002) split this process into six cyclic phases:

- *change capturing*, where the changes to be performed are identified;
- *change representation*, where these changes are formally represented;
- *semantics of change*, where the implications of these changes to the ontology are determined;
- *change implementation*, where the changes are applied to the ontology;
- *change propagation*, where the changes are propagated to dependent elements;
- *change validation*, where the ontology engineer reviews the changes, undoes them if needed and possibly identifies the need for more changes, reinitializing the cycle.

For the purposes of this thesis, the change capturing and change representation phases are of particular interest. I defined four types of change capturing partially based on Stojanovic & Motik (2002):

- *structure-driven*: analyses the structure of the ontology. It is based on heuristics such as a concept with a single subconcept may be merged with its subconcept, or if there are more than a dozen subconcepts for a concept, then an additional layer in the concept hierarchy may be necessary;
- *usage-driven*: considers the usage of the ontology to identify the interests of users in parts of ontologies. Parts of an ontology that are never used, may be considered outdated;

- *data-driven*: is based on changes to the underlying data set that was at the origin of the ontology;
- *instance-driven change discovery*: reveals implicit changes in the domain, that are reflected in ontology instances derived from techniques such as data-mining.

In the change representation phase, the necessary changes that were identified in the previous phase need to be represented in a suitable format. There are two main types of changes: elementary and composite. Elementary changes represent fine grained and simple changes, such as the deletion or addition of a single element (concept, property, relation) from the ontology. Composite changes represent more coarse grained alterations, and since they can be replaced by a series of elementary changes, below elementary changes will be referred to simply as changes.

Although Flouris *et al.* (2008) and Pinto *et al.* (1999) provide an exhaustive terminology for ontology change, some finer grained aspects of ontology evolution remain confusing, with several terms being used in an ambiguous fashion. For the purposes of this work, it becomes relevant to explicitly define and distinguish three related terms: ontology extension, ontology refinement and ontology enrichment. Although ontology extension is often used interchangeably with both refinement and enrichment, I define them as follows:

Definition 6. *Ontology extension is the process by which new single elements are added to an existing ontology.*

Thus, ontology extension is concerned with elementary changes of the addition type. Many reasons can motivate such a change, such as new discoveries, access to previously unavailable information sources, a change in the viewpoint or usage of the ontology, a change in the level of refinement of the ontology, etc, but they all rely on the finding of new knowledge. Ontology extension encompasses both ontology refinement and ontology enrichment.

Definition 7. *Ontology refinement is the addition of new concepts to an ontology, where a new subsumption relation is established between an existing concept and the new concept.*

Definition 8. *Ontology enrichment is the process by which non-taxonomical relations or other axioms are added to an existing ontology.*

Ontology extension is usually accomplished manually, however, in domains with a high rate of change such as biomedical research, ontologists struggle to keep up with the production of scientific knowledge. This makes the application of automated or semi-automated ontology extension techniques very desirable.

Chapter 3

Ontology Extension

The following sections give a general introduction and describe the state of the art in ontology extension approaches.

3.1 Ontology extension approaches

The most common data source used in automated ontology extension is natural text, due to its availability and coverage. However, other resources can also be used, including related ontologies and other vocabulary resources such as glossaries and thesauri. To handle these kinds of resources, as will be shown below, automated ontology extension can use techniques inherited from related areas of ontology engineering: ontology learning and ontology matching.

The identification of new concepts and relations from text typically resorts to the same kind of approaches as the task of learning new concepts in ontology learning. In fact, some semi-automated ontology learning systems actually require a seed or top ontology to be given a priori, which is then completed using automated techniques. The insertion of the new concepts at the appropriate position is usually addressed by machine learning techniques that classify the new concept into an existing ontology class.

Alternatively, ontologies can also be extended by the integration of concepts belonging to related ontologies, for which an alignment has been derived. In this scenario, if an equivalence is found between a concept from the master ontology and a concept from a related ontology, the master ontology can be extended with the subconcepts of its matching concept, or with concepts related to the matched concept through relation types present in the original master ontology, provided all constraints are maintained.

While most studies use only ontology learning techniques, some works employ both kinds of techniques, using ontology learning techniques to learn an ontology from text, and then ontology matching techniques to align it to the master ontology.

A schematic of the strategies used in ontology learning and ontology matching is provided, relating them to the techniques they employ, in Figure 3.1. This schema functions as an overview of the different approaches available to ontology extension.

3.2 Ontology learning strategies

Ontology learning is the process of automatically or semi-automatically building an ontology from a given corpus or data set (Cimiano *et al.*, 2004). Ontology learning approaches can be classified according to several dimensions, including data sources, units to be learned, learning targets, learning strategies, and learning techniques and knowledge support (Zhou, 2007). Ontologies can be learnt from various sources with various degrees of complexity and formalization, ranging from unstructured natural language text to semi-structured text (such as HTML and XML) and to highly structured data such as glossaries, database schemas and UML. However, most ontology learning methods are based on natural language processing (NLP) of relevant texts (e.g. articles, web pages), and several resort to machine learning techniques as well.

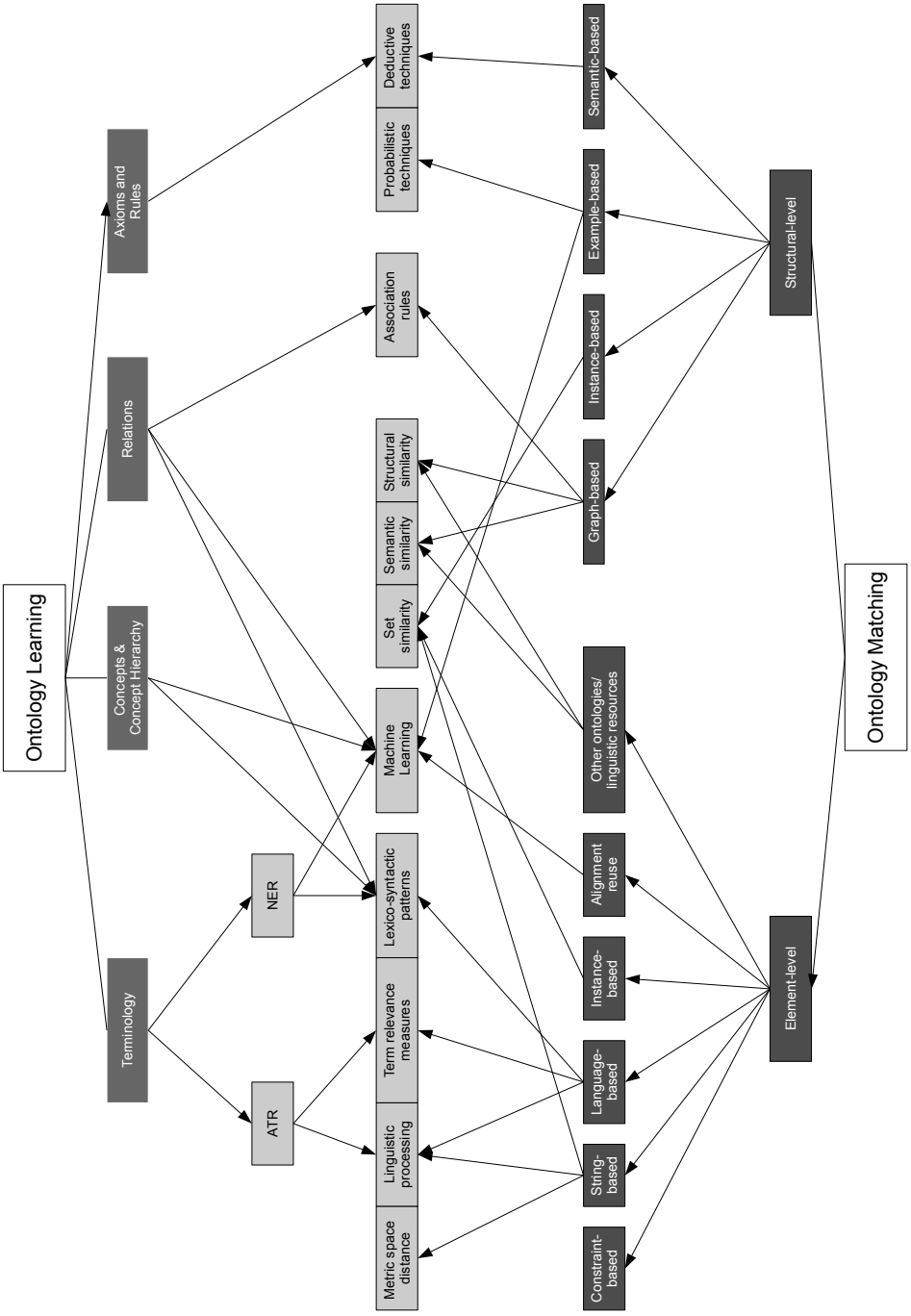

Figure 3.1: Techniques used in Ontology Learning and Ontology Matching, and the approaches that employ them.

Cimiano *et al.* (2004) proposed an ontology learning layer cake that organizes the different tasks that compose ontology learning, according to their dependencies. The layer diagram is composed of 8 levels, with the results of the lower tasks tipically being used as input for the higher tasks. Figure 3.2 depicts this structure and a simplification I propose to group the tasks into more general slices.

The two bottom levels correspond to the extraction of relevant terminology, which serve as the basis for the formation of concepts. Unlike terms, concepts are ontological entities, corresponding to an abstraction. The concept hierarchy layer is devoted to the identification of hyper/hyponymy relations between concepts, whereas the relation layer is dedicated to other kinds of relations between concepts. When these relations have been identified it is possible to deriving a hierarchy between them, which is a less commonly addressed task in ontology learning. The last two layers are the most challenging tasks in ontology learning due to the high level of complexity they can entail, however the adherence to a specific ontology language limits the types of allowed axioms.

Each of these tasks is addressed below, but for simplicity I group them in more general slices:

Cimiano *et al.* (2004) Structure	Proposed Structure
General Axioms	Axioms and Rules
Axiom Schemata	
Relations Hierarchy	Learning Relations
Relations	
Concept Hierarchy	Learning Concepts and their Hierarchy
Concept Formation	
Synonyms	Extracting relevant terminology
Terms	

Figure 3.2: Ontology learning layer cake by Cimiano *et al.* (2004) (left) and a proposed simplification (right).

3.2.1 Extracting relevant terminology

At every ontology's core is a terminology, thus, in ontology learning it is crucial to extract relevant terms that serve as the basic units of the ontology. The extraction of domain-specific terms from text is called term identification. It is a field which has been very active in the last two decades, mainly due to the increasing availability of electronic text and to the growing complexity of knowledge-based applications. Term identification is particularly important in the biomedical and healthcare domains, where a dinamycally changing terminology poses new challenges everyday. Term identification is commonly described as a three step task (Krauthammer & Nenadic, 2004):

1. **term recognition**, which marks the word or set of words that indicate the presence of a domain concept;
2. **term classification** (or categorization), which assigns terms to broad classes, e.g. in the biomedical domain, terms can be assigned to genes, proteins,diseases, etc;
3. **term mapping**, which links terms to well-defined concepts in an ontology or database.

Automatic term recognition in the biomedical domain is usually coupled with term classification, since it is more difficult to identify features that apply across a broad set of general terms than features that are specific to term classes. Term mapping is not performed as a part of ontology learning studies, since their goal is to derive ontology concepts from the identified terms. However, in the biomedical domain automatic term recognition crosses over with Named Entity Recognition (NER) since in biomedical terminology some named entities can also be seen as classes of entities (e.g. *alcohol dehydrogenase* can refer to a specific enzyme or to a class of enzymes).
Typically, automatic term recognition approaches involve three steps:

1. collecting candidate terms from text, extracting single or multiword terms from text;

2. ranking candidate terms, according to their relevance to the domain;
3. filtering candidate terms, according to a given threshold of relevance.

The collection of candidate terms usually focuses on the identification of noun phrases (NP), through the application of NLP techniques for normalization and linguistic processing such as part-of-speech tagging and tokenisation. It retrieves all possible terms in the from of single word or multi-word terms. The list of candidate terms is then ranked, according to the relevance and strength of the term in the pertinent domain, and weak/unrelated terms are filtered out. The ranking of terms can take into account distributional properties (statistical relevance), or contextual properties (contextual relevance).

Additionally, biomedical automatic term recognition can also employ NER approaches such as:

- dictionary-based, which use terminological resources to support the localization of a term in text
- rule-based, which rely on manually built rules, usually based on lexical, morphological and syntactical patterns to identify terms
- case-based, which leverage on annotated corpus to apply machine learning techniques to learn useful models for term recognition and classification or to support statistical approaches.

3.2.2 Learning Concepts and their Hierarchy

After the relevant terminology has been extracted, it is necessary to learn concepts and their hierarchy to build the backbone of the ontology: a taxonomy.

A concept can be defined intentionally (by a descriptive label or its relationships to other classes) or extensionally (by specifying a set of instances belonging to it) (Cimiano *et al.*, 2006). The assignment of a

description to a concept can be based on linguistic analysis, for instance Velardi *et al.* (2003) leverage on WordNet to describe compound terms based on their component's descriptions. The definition of a concept by its relations to other concepts addresses simultaneously the tasks of concept learning and concept hierarchy learning.

Following Maedche & Staab (2004), the extraction of taxonomic relations can be classified into three kinds of approaches:

- *statistics-based extraction using clustering*, which clusters terms based on distributional data, i.e. the frequency of the terms in a given context;
- *statistics-based extraction using classification*, which uses an already substantial hierarchy to learn the distributional representation of a training corpus and then classifies new terms according to the learned model;
- *lexico-syntactic pattern extraction*, which uses regular expressions that convey taxonomic relations to extract them.

3.2.3 Learning Relations

Once a taxonomic hierarchy is constructed, an ontology can be enriched by the addition of other types of relations between the concepts to model domain-specific properties. The task of relation extraction involves both the identification of anonymous relations between concepts and their labeling, and is considered one of the most challenging (Maedche & Staab, 2000) and least addressed (Sanchez & Moreno, 2008) in ontology learning, due to the complexity in ascertaining how many and what type of conceptual relationships should be modeled in a particular ontology. Furthermore, the assignment of labels to relations is also difficult since various relationships among instances of the same general concepts are possible (Maedche & Staab, 2002). Moreover, even if the semantic is clear, it might still be hard to guess which among several synonymous labels are preferred by a certain community (Kavalec & Svaték, 2005).

The learning of non-taxonomic relations within an ontology can be seen as a subfield of relation extraction from text, since the first is focused on establishing relations between the concepts of an ontology, while the second deals with the identification of relations between named entities, that may or may not correspond to ontology terms.

Methods for the extraction of non-taxonomic relations can be roughly split into three categories:

- statistical analysis based, such as co-occurrence analysis and association rules
- rule-based, which usually employ lexico-syntactic patterns
- machine learning based

Some systems integrate both statistical and rule-based approaches to discover relations, such as (Weichselbraun *et al.*, 2009)'s system that combines co-occurrence analysis, with patterns and WordNet queries to propose relations, which are then filtered by spreading activation.

3.2.4 Learning Axioms and Rules

Most ontology learning approaches do not concern themselves with learning complex ontologies. However, the success of OWL is giving rise to some advances in this area, given its ability to model expressive axiomatizations. Völker *et al.* (2008) propose two complementary approaches for learning expressive ontologies: one for generating formal class descriptions from natural language definitions extracted, e.g., from online glossaries and encyclopedias, and another that relies on a machine learning classifier for determining the disjointness of two classes. Haase & Völker (2005) generate consistent OWL ontologies from learned ontology models by taking the uncertainty of the knowledge into account.

3.2.5 Ontology Learning Systems

Several systems for ontology learning have been proposed in recent years. Table 3.1 describes the most relevant systems in terms of learning targets, data sources, learning techniques and external knowledge sources.

Ontology learning approaches can be classified according to several dimensions, including learning units (e.g. word, term), learning targets (e.g. concept, relation), data sources, strategies, techniques and type of knowledge support (Zhou, 2007).

3.3 Ontology Matching strategies

Ontology matching has been defined as "finding correspondences between semantically related entities of different ontologies" (Euzenat & Shvaiko, 2007). These correspondences are called alignments, and may represent not only equivalence, but also other kinds of relations, such as consequence, subsumption, or disjointness. The matching process can be seen as a function that determines the alignment A for a pair of ontologies o_1 and o_2. This function can have several input parameters, such as: the use of an initial alignment, matching parameters (e.g. weights, thresholds), and external resources.

Definition 9. *The matching process can be seen as a function f which, from a pair of ontologies to match o_1 and o_2, an input alignment A, a set of parameters p and a set of oracles and resources r, returns an alignment A' between these ontologies: $A' = f(o_1, o_2, A, p, r)$*

There are three areas in which users can be involved in a matching solution: (1) by providing initial alignments (and parameters) to the matchers, (2) by dynamically combining matchers, and (3) by providing feedback to the matchers in order for them to adapt their results.

System	Learning targets	Strategies	External knowledge
TextToOnto Text2Onto (Cimiano & Völker, 2005)	terminology hierarchy relations other axioms	statistical relevance pattern extraction contextual relevance	
OntoLearn (Navigli & Velardi, 2004)	terminology synonyms hierarchy relations	statistical relevance pattern extraction	WordNet
OntoLT (Buitelaar *et al.*, 2004) RelExt (Schutz & Buitelaar, 2005)	terminology hierarchy relations	statistical relevance rules	user defined rules
ASIUM (Faure & Nédellec, 1998) Mo'K (Bisson *et al.*, 2000)	hierarchy relations	syntactical patterns conceptual clustering	
HASTI (Shamsfard, 2004)	terminology hierarchy relations other axioms	syntactical morphological pattern extraction clustering	seed ontology

Table 3.1: Ontology Learning systems

Matchers can be elementary (individual) or compositional. Elementary matchers are addressed first, followed by several matching strategies that include the composition of matchers, global matching strategies and user involvement.

3.3.1 Elementary matchers

Following a classification proposed by (Euzenat & Shvaiko, 2007), matching techniques can be classified according to their granularity (element-level vs. structure-level) and then according to how they interpret the input information (syntactic vs. external vs. semantic).

3.3.2 Element-level techniques

Element-level matching techniques compute correspondences by analyzing entities or instances of those entities in isolation, ignoring their relations with other entities or their instances. These can use internal knowledge only, i.e. information contained in the ontology itself, or incorporate external knowledge in the form of reusable alignments, upper or domain ontologies and other linguistic resources. Internal knowledge-based methods include:

- *string-based methods* which use linguistic processing, to reduce strings to a common format and then process them according to metric space distance or similarity measures such as Levenshtein or Manhattan block-distance. These can be further enhanced by term relevance measures (see section **??**)
- *language-based methods* also use linguistic processing methods, but go beyond string based methods to analyze lexico-syntactic patterns and apply term recognition methods
- *constraint-based methods* are based on the comparison of the internal structure of entities, such as types, range, cardinality or multiplicity of properties, and keys, and are usually used in combination with other techniques.

- *instance-based methods* rely on the notion is that if two ontology entities share the same or similar sets of instances, then they can represent a match.

External knowledge-based methods include:

- alignment reuse-based methods
- other ontologies or linguistic resources-based methods

3.3.3 Structure-level techniques

Structure-level techniques compare ontology entities or their instances based on their internal relations with other entities or their instances. These can also use external knowledge, for instance in the form of instances, when these are not part of the ontology, or in the form of previous alignments. Several types of methods can be employed at the structural level:

- *graph-based methods* which consider entities similar if their neighbors are also similar, using structural or semantic similarity
- *instance-based methods* which consider entities similar if their instances are also similar using set similarity measures or formal concept analysis
- *example-based methods* which leverage on examples of correct and incorrect alignments and apply machine learning or probabilistic techniques to derive the matches.
- *semantic-based methods* which use deductive techniques to arrive at matches.

3.3.4 Global similarity computation

The similarity between two ontology entities may involve the ontologies as a whole, so that the final similarity between two entities may ultimately depend on all of them. Similarity flooding (Melnik *et al.*, 2001) defines the propagation factors according to the outer degree of

the node that represents the entity, i.e., the number of adjacent entities. OLA (Euzenat *et al.*, 2004) computes several alignments with different combinations of normalized weights, and uses the one that generated the best alignment as a propagation factor. DSI and SSC (Cruz & Sunna, 2008a) define the propagation factors according to a constant percentage (main contribution percentage) that represents the fraction of the element-level similarity used to determine the overall similarity. Anchor-Flood (Seddiqui & Aono, 2009) uses the similarity between neighbors of a proposed matching pair to support it.

3.3.5 Composition of matchers

After the similarities between ontology entities have been computed, it is necessary to employ more global strategies to arrive at a final alignment. These include the aggregation of the elementary techniques discussed previously, global similarity techniques (see above) and alignment extraction.

The aggregation of several distinct matches has been shown to improve matching results (Cruz & Sunna, 2008a; Lambrix & Tan, 2006). There are two basic ways of combining several matching techniques:

- sequential composition, where the results of one matcher are fed to the next
- parallel composition, where distinct matchers are run concomitantly, and their results are combined following specific criteria,
  - homogeneous, in which the different kinds of data are processed by appropriate matchers
  - heterogeneous, in which the same input is used by competing matchers

Alignment extraction employs a series of techniques over matches between ontology elements, to derive a final optimized alignment for the full ontologies. These techniques can include trimming, which applies

thresholds to ensure only the best matches are considered; or maximal weight matching (or weaker variants like stable marriage) which optimize the global similarity.

3.3.6 Ontology Matching Systems

3.3.7 Alignment of biomedical ontologies

In 2007, the OAEI proposed as a task the alignment of the NCI Thesaurus describing the human anatomy and the Adult Mouse Anatomical Dictionary, which has been developed as part of the Mouse Gene Expression Database project. Both ontologies are part of the Open Biomedical Ontologies (OBO)[2]. The alignment of these ontologies can be evaluated through qualitative measure because there is a reference alignment for these ontologies.

Given that many of the alignments are rather trivial and can be found by simple string comparison techniques, the OAEI team created a measure of recall, recall+, which ignores the trivial matches, to distinguish the ability of the systems to identify non-trivial matches. In fact, the most interesting challenge in aligning these ontologies in an automated fashion lies in identifying these non-trivial correspondences, since without them the alignment has a limited usefulness.

The results of OAEI 2007 showed that systems using background knowledge, such as AOAS (Zhang & Bodenreider, 2007), SAMBO (Lambrix & Tan, 2006) and ASMOV (Jean-Mary *et al.*, 2009), generated the best alignments (Isaac *et al.*, 2007). These three systems employed the UMLS (Unified Medical Language System) as the domain knowledge source to support lexical matching of concepts. These systems also aggregated several matchers, mostly based on structural and terminological similarities, clearly indicating the possibility of combining the strengths of both domain specific and domain independent systems, to create a hybrid matching strategy. In 2008, the results did

[2]www.obofoundry.org

System	Element-level strategies	External resources	Structure-level strategies	Composition
Anchor-Flood	string, ling. resource, constraint	WordNet	graph, similarity propagation	
AMaker	string, language, constraint	WordNet	similarity propagation	linear weighted combination
AROMA	string, instance	document collection	graph	
ASMOV	string, ling. resource, constraint	common thesaurus	graph, instance	weighted average
CIDER	string, ling. resource	WordNet, web	graph	weighted sum
DSSim	upper-level ontologies	WordNet		fuzzy voting
GeRoMe	instance, string		similarity propagation,	weighted aggregation
HMatch	language, constraint	common thesaurus	graph	
kosimap	string, constraint		graph, semantic	weighted sum
Lily	string, constraint		similarity propagation, semantic	weighted sum
MapPSO	string, language, constraint	WordNet	graph	ordered weighted average
RiMOM	string	WordNet	graph, similarity propagation	linear -interpolation
SAMBO SAMBO dtf	string, ling. resource	common thesaurus WordNet	graph, example	weighted sum
SOBOM	string		similarity propagation, relations	sequential
SPIDER[1]	see CIDER	online ontologies	see CIDER graph	see CIDER
TaxoMap	language		graph	

Table 3.2: Ontology matching systems that participated in OAEI 2008 and 2009

not portrait this clear advantage of domain knowledge based methods (Caracciolo *et al.*, 2008). The systems that used background knowledge (SAMBO and ASMOV) produced F-measures of 0.85 and 0.71, respectively. While among the systems that do not rely on background knowledge, RiMOM (Li *et al.*, 2009) obtained the best result with an F-measure of 0.82. In OAEI 2009 (Ferrara *et al.*, 2009), the best system was SOBOM, with an F-measure of 0.855 and a recall+ of 0.431, which is below the previous year best recall+ of SAMBO, 0.586. AgreementMaker (Cruz & Sunna, 2008a) was a close second, with 0.831 of F-measure and a better recall+ of 0.489. It is interesting to note that neither of these systems use domain knowledge, but SAMBO, the best system in 2008 did not compete in 2009. Table 3.2 summarizes the results of task1 in the anatomy track for 2007, 2008 and 2009.

Other recent efforts in this area include the alignment of OBO (Open Biomedical Ontologies) disease ontologies using syntactic patterns and the UMLS semantics (Marquet *et al.*, 2007), the alignment of Gene Ontology biological process ontology to pathway ontology using artificial neural networks to define the weights of different similarities (Huang *et al.*, 2008) and the application of SAMBO to the alignment of a part of GO to a part of SigO (Signal Ontology) (Lambrix & Tan, 2006).

3.4 Ontology extension systems

This section describes relevant ontology extension systems of the last decade. There are several other systems that can be used in ontology extension, but here the focus is on systems that were devised with that application in mind, rather than other ontology engineering tasks. Table 3.3 summarizes the following descriptions.

Agirre *et al.* (2000) propose a system that uses the web to extract words related to WordNet concepts. This system first retrieves all web documents related to an ontology concept, using carefully constructed queries that try to discard documents that belong to different word

senses. These documents are then processed to yield a topic signature, a vector of the words in them and their frequencies, which are filtered to include only the words that appear distinctively in the collection of one ontology concept in respect to the others. The authors also used document clustering techniques to cluster WordNet concepts, since each concept was linked to a collection of documents, tackling the word sense proliferation in WordNet. The system was evaluated in word sense disambiguation tasks, and outperformed the precision of WordNet synonyms.

Velardi *et al.* (2001) have developed an ontology learning system, OntoLearn, that also functions as an ontology extension system since it links the generated hierarchies of concepts to a core ontology, the WordNet. This system uses a semantic disambiguation algorithm SSI (Navigli, 2005) to identify the correct concept in WordNet that corresponds to each root of the generated hierarchies. SSI uses the other concepts in the hierarchy as the context for disambiguation. Manual evaluation resulted in 85% precision and 53% recall.

Alfonseca & Manandhar (2002) developed a system that classifies unknown concepts into WordNet based on the similarity between the concept and WordNet synsets. It performs a top-down search, comparing the unknown concept to the most general synset and its immediate hyponyms. If the unknown concept is more similar to the general synset the search stops if it is more similar to any of the hyponyms the search continues. This system uses a semantic distance based on a topic signatures. A topic signature of a term t is the list of terms that co-occur with it and their frequencies or weights. The topic signatures are acquired automatically using the web following (Agirre *et al.*, 2000). This resulted in an accuracy of 13%. To further support decision, the choice of the appropriate synset is backed by hypernymy patterns, which increase the weight of synsets with consistent hyponyms. This increased accuracy to 28%. In another work, these authors extended their system to include subject, object and modifier signatures for a concept c based on the verbs where c is a subject, the

verbs and prepositions where c is an argument, and the adjectives and determiners that modify c respectively.

Faatz & Steinmetz (2002) proposed a method for ontology extension based on texts extracted from the web. The method assumes that candidate terms have already been identified and focuses on identifying which are the most relevant candidates , and where do they fit in the ontology. It is based on statistical information on word usage. It defines concepts as vectors of their collocators (words that occur near the concept in a piece of text), and uses similarity between ontology entities to derive similarity between collocators and ontology entities. If this similarity is above a given threshold, the term is proposed as an extension.

Pekar & Staab (2002) applied classification techniques to distributional data to classify a new term into an existing concept hierarchy. They proposed a tree ascending classification algorithm which extends the kNN method by making use of the taxonomic similarity between nearest neighbors. This algorithm was tested against standard classifiers and resulted in precision values around 15%.

Widdows (2003) uses a version of latent semantic analysis, where words are represented by vectors of their co-occurrence frequency with 1000 frequent words. These vectors constitute a matrix, or a Vector Space Model (VSM), whose dimensions are reduced. Similarity is then calculated using the cosine similarity measure. These terms are then added to WordNet based on the notion that the class of a set of terms is the hypernym that subsumes as many as possible of the members of that set. This is translated into an affinity score function that rewards hypernyms closer to the term, and penalizes hypernyms that do not subsume the term. Evaluation on WordNet reconstruction yielded a precision of about 80% for common nouns, 34% for proper nouns and 65% for verbs.

Witschel (2005) proposed a method that identifies new terms in large corpora using linguistic patterns. These terms are then described in

terms of the words that co-occur with them more often, which allows to compute the similarity between two terms via the number of words in their descriptions they share. These descriptions are used as features for a decision tree approach, where new terms are inserted into the hierarchy based on their similarity to the descriptions of the terms in the ontology. The hierarchy is traversed in a top-down approach, and the search stops when the average of the similarity values between the new concept and all the children of a given term at a given level is greater than the variance. This work bears some similarities to (Alfonseca & Manandhar, 2002) and its evaluation resulted in comparable values of precision around 10-15%.

Ruiz-Casado *et al.* (2005) developed a method for calculating the semantic distance between two terms that is based on the distributional hypothesis. This method, context-window overlapping, overcomes some of the limitations of vector space models, including specification of context and word order. It is based on the notion that the similarity between two terms can be given by the percentage of contexts of t_1 where t_2 can be used instead of t_1 to obtain a context of t_2, and vice-versa. The context of a term is defined as a narrow window of restricted length, and the Internet is used to collect them. This similarity metric was applied to the extension of WordNet using a top-down search algorithm, where a term and its children are considered candidates, based on the notion that a term in a sentence can be substituted by any of its hypernyms.

Liu *et al.* (2005) developed a system to semi-automatically extend ontologies based on textual data retrieved from the Web. This system uses ontology terms as seeds for a Lexical Analyzer that discovers candidate terms. The analyzer has 4 components: 1) co-occurrence analysis, whereby terms that co-occur at sentence and document level are ranked using log likelihood; 2)WordNet hyponyms, hypernyms and synonyms, based on matching the seed term to a WordNet term using vector space models; 3) Trigger phrase analysis, which uses regular expressions to find synonyms or hyponyms; and 4) Head noun analysis,

that extracts hypernyms from compound names. The candidate terms obtained are used to generate a network, where the links that connect them to the seed terms are weighted according to the trustworthiness of each type of analysis. Spreading activation is run over this network to generate the list of the most promising candidate terms to add to the ontology.

Mahn & Biemann (2005) presented a method that uses higher-order co-occurrences to generate ontology extension candidates. To generate higher-order co-occurrences for a term t, the N highest ranked co-occurrences of t are added to t as a pseudosentence and the co-occurrence calculation is applied to this extended corpus. Each iteration of this procedure generates a higher order of co-occurrences. The 2^{nd} and 3^{rd} order of co-occurrence were found to produce the best results.

Lee *et al.* (2006) proposed a system that automatically generated candidate terms for the Gene Ontology based on syntactic relations between existing terms. The candidate terms are more detailed concepts that are created by combination of conceptual units of hypernym terms. For instance, if a term t is composed of two conceptual units (that is, it results of the composition of two other terms in the ontology), such as 'chemokine binding', then new terms can be generating by substituting the conceptual units by their children, resulting for instance in 'C-C chemokine binding'. These candidate terms are validated by their presence in biomedical literature.

Novácek *et al.* (2008) proposed a system for the dynamic integration of automatically extracted knowledge from unstructured sources into manually maintained formal ontologies. The system integrates learned ontologies using Text2Onto into a master ontology, using matching techniques from the ontology alignment API developed by INRIA Rhone-Alpes and negotiation techniques from Laera *et al.* (2006). This system was applied to the extension of a fragment of the Gene Ontology cellular component branch using Wikipedia entries, with a precision ranging from 50% to 85% depending on algorithm iterations.

Bendaoud *et al.* (2008) developed a system, PACTOLE, that semi-automatically extends an ontology using a collection of texts. It builds two concept hierarchies using formal concept analysis, one based on a collection of texts, and the other based on the initial ontology. The text analysis is based on syntactic dependencies between the named entities and properties, to uncover pairs (object, property) that are subsequently manually reviewed. Formal concept analysis is applied to these pairs to generate a concept lattice. The other concept lattice is directly derived from the initial ontology hierarchy. The two lattices are then merged and the final lattice is transformed into an ontology. PACTOLE was applied to the domain of astronomy, and the similarity between concepts from the initial ontology and concepts extracted from text was evaluated, yielding a precision of 75% and a recall of 30%.

Jimeno-Yepes *et al.* (2009) proposed an algorithm for ontology extension that is based on feedback from information retrieval processes supported by ontologies. It is based on the assumption that there is a set of terms which if added to the ontology supporting the query formulation will have a positive impact on precision and/or recall. The system used term extraction using a shallow parser or the Whatizit system (Rebholz-Schuhmann *et al.*, 2008) for NER, taxonomic relations are extracted based on Hearst patterns and non-taxonomic relations rely on co-occurrence analysis and sentence categorization (where SVM were the best performers). This algorithm was tested on two corpus, PGN-disease association and protein-protein interaction for yeast: the first clearly improved the performance of the IE task (30%), doubling the baseline; while the second achieved comparable results to the baseline.

Wächter *et al.* (2010) developed a system for the creation and extension of ontologies by semi-automatically generating terms, definitions and parent-child relations from text. It generates terms by identifying statistically significant noun phrases in text and uses pattern-based web searches to find definitions and parent-child relations. This system

can retrieve up to 78% of definitions and up to 54% of child-ancestor relations.

3.4.1 Issues in ontology extension systems

The performance of most of the described systems is still lacking, highlighting the difficulties in this area. However, one of the best performing systems (Novácek *et al.*, 2008) reached precision values of 50 to 85% when extending a portion of the Gene Ontology. These higher precision values obtained can be explained by two facts: 1) it uses Wikipedia as a corpus, which is an easier corpus to explore given its structured contents and 2) the domain of the portion of the ontology to extend is a common one, ensuring that there are several entries in Wikipedia related to it. In fact, although precision values are good, the authors themselves state that most suggestions are rather simple and obvious. Wikipedia and other similar resources can be valuable sources for extension of ontologies with common domains, however when the ontology to extend reaches a high level of specificity, the only possible source of knowledge is scientific literature where the needed level of detail is present. Nevertheless this remains most interesting approach, and it motivated some of the strategies in this thesis.

One common feature to most systems is that they are based on manually constructed corpora. These can provide the level of detail needed to extend very specific domains, however building them can become a bottleneck in the extension strategy given the time and expertise needed to identify the relevant documents. It is arguable that the quality of the corpora can have quite an impact on the methods performance, since its relevance to the domain is crucial. Moreover, when extending large ontologies with broad domains, using a very large and generic corpora can hinder the task of identifying the relevant terms given the profusion of existing terms.

These issues need to be addressed when contemplating the extension of

System	Ontology Learning Techniques	Ontology Matching Techniques	Sources	Ontology	Performance
Agirre *et al.* (2000)	term relevance clustering	NA	web	WordNet	NA
Velardi *et al.* (2001)	OntoLearn contextual relevance	NA	web	WordNet	85% precision 35% recall
Alfonseca & Manandhar (2002)	classification hypernym patterns	NA	web	WordNet	28% accuracy
Faatz & Steinmetz (2002)	term relevance set similarity	NA	text	k-med[a]	NA
Pekar & Staab (2002)	classification semantic similarity	NA	text	WordNet	15% precision
Widdows (2003)	co-occurrence vector similarity	NA	text	WordNet	35-80% precision
Witschel (2005)	pattern extraction classification	NA	text	GermaNet	10-15% precision
Ruiz-Casado *et al.* (2005)	distributional properties contextual relevance	NA	text	WordNet	NA
Liu *et al.* (2005)	distributional properties matching to WordNet pattern extraction morpho-syntactic analysis	NA	web	climate change ontology	NA
Mahn & Biemann (2005)	co-occurrence	NA	text	WordNet	NA
Lee *et al.* (2006)	syntactical analysis	NA	ontology vocabulary	Gene Ontology	NA
Novácek *et al.* (2008)	Text2Onto	INRIA alignment API	Wikipedia	Gene Ontology	50-85% precision
Bendaoud *et al.* (2008)	syntactical patterns	structural similarity	text	astronomy ontology	NA
Jimeno-Yepes *et al.* (2009)	patterns co-occurrence	NA	text	UMLS[b]	NA
Wächter *et al.* (2010)	term relevance patterns	NA	text web	GO, MeSH, OBO and UMLS	13-80% of terms are similar to concepts

Table 3.3: Ontology extension systems

[a] www.k-med.org/

[b] Unified Medical Language system www.nlm.nih.gov/research/umls/

biomedical ontologies which often combine very board but very specific domains.

Chapter 4

Basic Concepts

4.1 Machine Learning

A generally accepted definition of machine learning was given by Mitchell (1997):

> A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E.

This experience, or empirical data, is used by the machine learning algorithm to derive a model capable of capturing the underlying probability distribution of a set of examples. The majority of machine learning algorithm fall in one of two categories: supervised and unsupervised learning. In supervised learning tasks, such as classification, training data labels are used to learn am model capable of labeling new input data. In unsupervised learning, the data is unlabeled and the purpose of the learning algorithm is to detect the data's internal structure. Clustering is a common example of unsupervised learning. The remainder of this section concentrates on supervised learning given its relevance to this thesis.

Training examples in classification tasks are commonly represented as vectors of feature values. A classification model is then a function that

assigns a label or class to a vector representation of an example.

The evaluation of classification results is usually based on a set of widely accepted measures:precision, recall and f-measure. In 2-class problems, these measures are calculated based on a confusion matrix:

predicted class	correct class	
	true	false
true	true positive	false positive
false	false negative	true negative

Precision is the number of correctly classified examples in relation to the total number of examples classified as true.

$$precision = \frac{true\ positives}{true\ positives + false\ positives} \tag{4.1}$$

Recall is the number of correctly classified examples in relation to the total number of true examples.

$$recall = \frac{true\ positives}{true\ positives + false\ negatives} \tag{4.2}$$

F-measure balances precision and recall by calculating their harmonic mean.

$$f - measure = 2 \times \frac{precision \times recall}{precision + recall} \tag{4.3}$$

4.2 Semantic Similarity

A knowledge-based semantic similarity measure can be defined as a function that, given two ontology concepts or two sets of concepts annotating two entities, returns a numerical value reflecting the closeness in meaning between them(Pesquita *et al.*, 2009b). There are essentially two types of approaches for comparing terms in a graph-structured ontology such as GO: edge-based, which use the edges and their types as the data source; and node-based, in which the main data sources are the nodes and their properties. Edge-based approaches are mainly

based on counting the number of edges in the path between two terms (Rada *et al.*, 1989). The most common technique, distance, selects either the shortest path or the average of all paths, when more than one path exists. This technique yields a measure of the distance between two terms, which can be easily converted into a similarity measure. While these approaches are intuitive, they are based on two assumptions that are seldom true in biological ontologies: (1) nodes and edges are uniformly distributed, and (2) edges at the same level in the ontology correspond to the same semantic distance between terms.
Node-based approaches do not suffer from these issues since they rely on node (concept) properties. One concept commonly used in these approaches is information content (IC), which gives a measure of how specific and informative a concept is. The IC of a concept c can be quantified as the negative log likelihood,

$$-\log p(c)$$

where $p(c)$ is the probability of occurrence of c in a specific corpus (such as the UniProt Knowledgebase), being normally estimated by its frequency of annotation. Alternatively, the IC can also be calculated from the number of children a term has in the GO structure [7], although this approach is less commonly used (Seco *et al.*, 2004). The concept of IC can be applied to the common ancestors two terms have, to quantify the information they share and thus measure their semantic similarity.

Part II

Methods for Semi-Automated Biomedical Ontology Extension

Chapter 5

A Framework for the Semi-Automated Extension of Biomedical Ontologies

A framework for semi-automated ontology extension within the context of biomedical ontologies must take into account the specific challenges of the domain as well as its opportunities.

5.1 Challenges and opportunities in extending a biomedical ontology

Most ontology extension studies focus on adding new nouns to WordNet, a lexical resource that organizes English nouns into a hierarchy. Although many of the issues faced by these studies are the same ones that need to be considered when extending bio-ontologies, there are some crucial differences:

- Bio-ontologies have well-defined domains (e.g., protein functions, gene sequence features, anatomy), which means that the techniques used to extract new terms from text need to be able to differentiate between general terms and domain terms

- Biomedical terminology is complex and ambiguous.
- Bio-ontologies typically have a low level of axiomatization, with few types of relations defined between the concepts.
- Biomedical ontologies are usually quite large with many thousands of concepts.
- Biomedical ontologies have a faster rate of evolution, since biomedical knowledge changes faster than a language.

However, the biomedical domain also presents some interesting characteristics that can be explored in the context of ontology extension:

- There are many biomedical ontologies with overlapping domains
- Biomedical ontologies are terminologically rich
- There is an abundance of scientific literature indexed in PubMed

5.2 Framework

Building upon the identified challenges and opportunities I designed the framework described in Figure 5.1.

This framework has three main components:

Change Capturing component is in charge of change capturing and should be able to identify the areas of the ontology in need of extension. This component is responsible for tackling the issues related to the large amounts of available scientific literature and related ontologies.

Matching component where an ontology matching method is used to align the areas of the ontology to extend to other relevant ontologies and thus support ontology reuse.

Learning component which is responsible for taking text corpora built based on the results from the Change Capturing component and extracting novel ontology concepts.

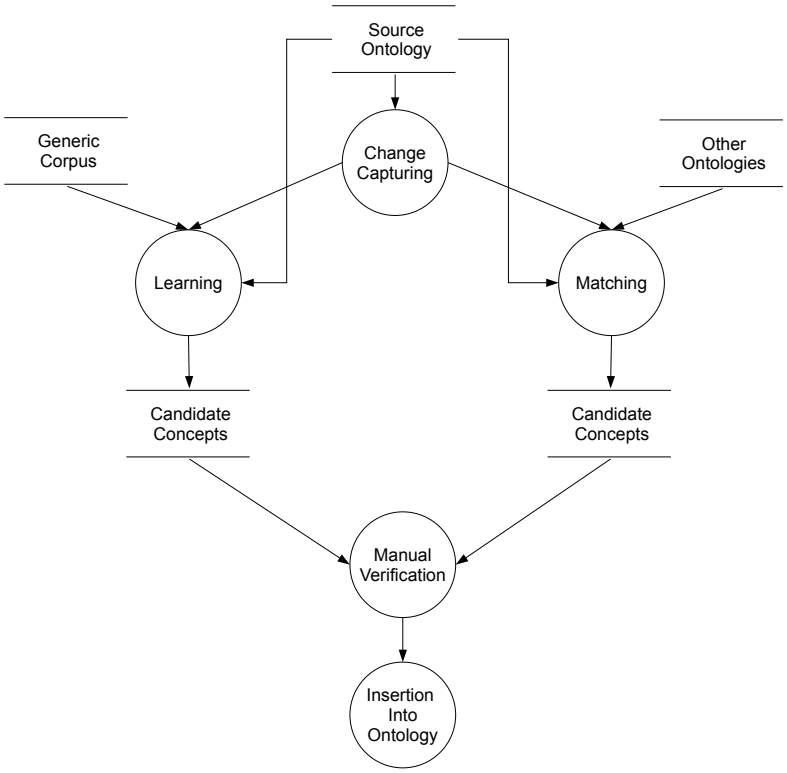

Figure 5.1: Data flow diagram of the proposed framework.

The Change Capturing component functions as the first step in automated ontology learning or extension systems. Ontology learning systems usually rely on the analysis of a manually constructed corpus of documents pertaining to the domain of interest and their performance is closely coupled to the relevance of these documents. The challenge of focusing the ontology given an heterogenous corpus in ontology learning has been identified (Brewster *et al.*, 2009a). This challenge is amplified when it comes to ontology extension of large ontologies, as is the case of many biomedical ontologies. A comprehensive corpus for these ontologies would be quite large and building and then processing it would be cumbersome. By applying the proposed strategy, ontology developers can identify subdomains to extend, cre-

ate tailored corpus for them, and then run the learning systems over them, reducing the amount of data they have to process to identify new concepts. Likewise, when using ontology matching as an integral part of ontology extension, identifying the areas to extend can be used to narrow down on specific ontologies to match, to support the integration of elements from other ontologies.

The Matching and Learning components are independent and can be run in parallel, offering complementary functions. The methodology employed within the Matching component should be specifically tailored to the biomedical domain, and thus should be able to take advantage of the rich terminology and to handle large ontologies. The methodology used in the Learning component should be able to filter out general concepts in favor of domain ones without being hindered by the complex and ambiguous terminology of this domain. Whereas several methods and systems exist in the areas of ontology matching and learning, there is none in automated change capturing. Consequently, in this thesis there is a focus on developing methods for the Change Capturing component, whereas work in other components is more focused on improving and adapting existing methods to the specificities of the biomedical domain.

The Manual Verification component corresponds to the manual analysis of the candidate concepts by an expert before their integration into the ontology. By generating the list of candidate concepts in an automated fashion, the main contribution of this framework for ontology developers lies in the speeding of the process of extension, thus releasing the experts to focus on more complex ontology evolution issues.

5.3 Analyzing Ontology Extension

An analysis of ontology extension should by definition, focus on both refinement and enrichment, and analyze several versions of the same ontology during a time period. The decision on the time period to analyze should be based on the age of the ontology as well as the availability and frequency of new version releases.

Three key aspects are proposed for analyzing ontology refinement:

1. depth of new classes, i.e. minimum distance to the root class over *is_a* and *part_of* relations.
2. number of new classes that are children of existing *vs.* newly added classes
3. number of new classes that are children of leaf classes

The first and third aspects capture the general direction of the refinement of the ontology, where additions at a greater depth and to leaf classes represent vertical extension whereas additions at middle depth and to non-leaf classes represent horizontal extension. These aspects are helpful to analyze the level of detail and coverage provided by the refinement. The second aspect is related to another interesting characteristic of refinement, whether new classes are inserted individually or whether as part of a new branch.

The following are proposed for analyzing ontology enrichment:

4. age and depth of the classes linked by the new relation (i.e. whether the relation is established between old classes, between an old and a new class or between new classes)

This aspect is intended to capture first at what level of specificity do the enrichment events take place, and secondly if enrichment happens alongside refinement or if it succeeds it.

5.3.1 Analyzing the Gene Ontology Extension

Based on the aspects identified in the previous section, twelve versions of the Gene Ontology and its annotations were analyzed. These covered a period of 6 years from 2005 to 2010 and were six monts apart or as close to that as possible (See the first twelve versions described in Table 6.2. Using these six month intervals, new classes represent about 5% of all classes in the ontology. In the context of GO, enrichment corresponds to the insertion of new non *is_a* relations between existing or newly inserted classes.

Figures 5.2, 5.3, 5.4 and 5.5 show the results of the analysis of each aspect. In all three hierarchies, the majority of new subclasses are added as children of non-leaf classes, resulting in a prevalence of horizontal extension. Also, the refinement of molecular function and cellular component occurs mostly via single insertions, whereas in the biological process groups of related classes are inserted together. Regarding enrichment, in biological process, a considerable portion of relations are established between two newly inserted classes, whereas in cellular component, the majority is made between an existing and a new class.

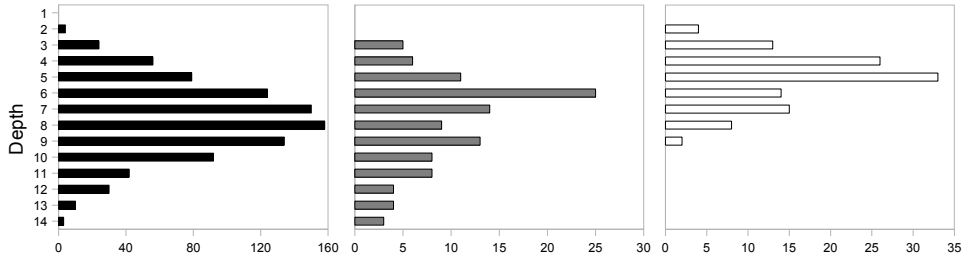

Figure 5.2: Average depth of new classes

The application of the proposed conceptual framework for ontology extension analysis to GO has yielded some interesting results. Firstly, the majority of new classes are not added to leaf classes, resulting in a horizontal growth of the ontology. This means that GO is not adding increasingly specific classes but rather fleshing out. Secondly, it was shown that in GO refinement happens by two major modes: individual

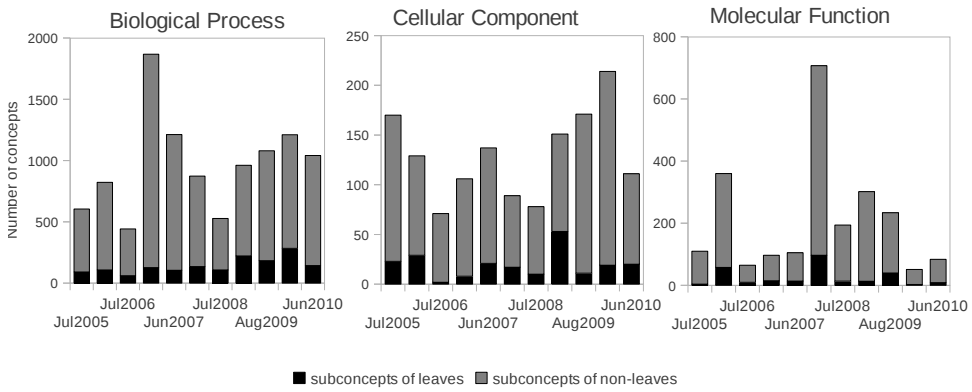

Figure 5.3: Ancestry of new classes (leaves or non-leaves) by ontology version

insertions and group insertions. The first occurs frequently in all GO hierarchies, whereas the second is only common in the biological process hierarchy. This is in line with the fact that most of GO's special interest groups belong to the biological process area and their work is more focused on modelling portions of their areas of interest rather than making individual insertions. This refinement by branches in the biological process hierarchy is also captured by the enrichment analysis, where there is a high proportion of new enrichment relations that are established between new classes. One issue with using path-based depth to define the sub-graphs of GO that are subject to extension, is that it can cause bias, since terms at the same depth do not necessarily express the same degree of specificity Alterovitz *et al.* (2010). However, this issue is outweighed by the need to create sub-graphs independently of their number of annotations, so as not to introduce a bias to the annotation based rules.

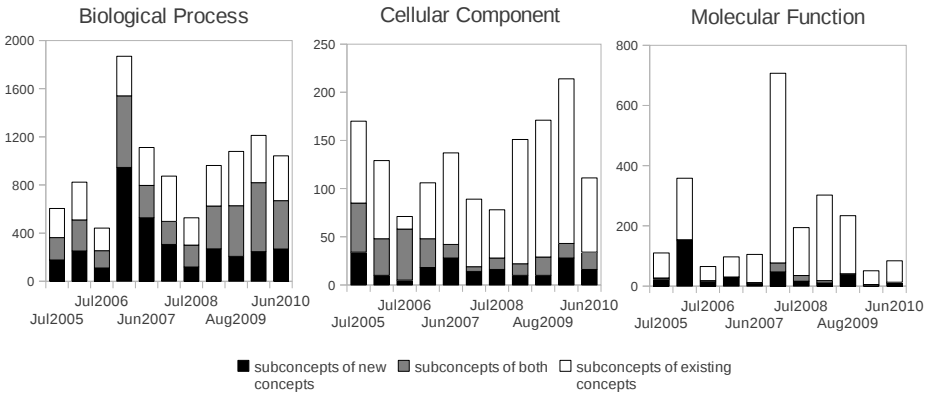

Figure 5.4: Ancestry of new classes (existing or new parents) by ontology version

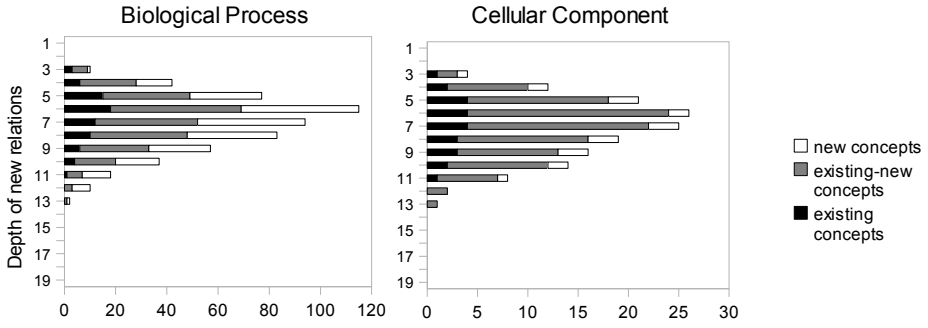

Figure 5.5: Depth and age of the classes in new enrichment relations.

Molecular Function not shown, since it contains less than 10 non *is_a* relations.

Chapter 6

Prediction of Ontology Extension

6.1 Introduction

Scientific literature remains the principal vehicle for scientific knowledge record and communication, which makes literature analysis a large part of the mandatory work needed for ontology evolution. However, the rate at which new papers are published prevents a timely manual analysis of all relevant documents. For instance, the general domain of the most successful bio-ontology, the Gene Ontology, is protein function. A query for *protein AND function* in PubMed, retrieves 3.3 million publications. While this number is certainly an underestimation of the number of publications within the domain of GO, it still means that on average, there are 500 new scientific papers per day on this subject. An important step in the automation of ontology evolution is then the automation of text analysis to retrieve pertinent information, i.e. text mining. But despite text mining of biomedical literature being a thriving field (Ananiadou & Mcnaught, 2006), its application to bio-ontology evolution is a much less active area. In fact, ontology extension, and its sister area ontology learning remain largely unexplored in a biomedical context, with few works reported

(Brewster *et al.*, 2009b).

Most ontology learning and ontology extension systems depend on a corpus of domain texts which are used to extract the relevant terminology and relations. The text corpus is given a priori to the systems, and is usually composed of documents such as full journal articles on the general domain (Brewster *et al.*, 2009b), benchmark text collections (e.g. Reuters RCV1 (Fortuna *et al.*, 2006)), or website pages (Cimiano & Staab, 2005). This means that somewhere in this process these collections had to be manually selected according to some criteria (e.g. availability, quality, etc). However, if we are truly to automate ontology extension, this collection should also be automatically obtained.

A simple approach to generate a document collection, common in the biomedical domain, is to query PubMed with keywords pertaining to the domain in question and retrieve the returned abstracts. When the domain is fairly small this strategy may be quite successful, but when the domain is large and complex, some issues arise. Retrieving all documents related to a domain may generate a very large collection of documents. However, many of these publications will not contain any new information pertinent to the extension of the ontology.

To filter out irrelevant documents, we can either set limits in time or in scope. Setting a time limit would have to rely on the assumption that the current state of the ontology reflects all available knowledge at a given point in time, so we can disregard any publications before that date. Thus, choosing such a point in time would have to be based on common sense rather than on any proof. Taking again the example above, a time limit of two years would result in a little over 340 thousand documents, a much more manageable dataset.

Filtering based on scope relies on a distinct strategy to generate the document collection: instead of a single search for keywords relating to the full domain, several queries are used, each based on a distinct subdomain within the ontology to extend. This strategy opens up an interesting opportunity whereby the document collection can be tailored to specific areas within the ontology. Consequently, a document

collection can be specifically designed to cover ontology areas that need to be updated.

However, identifying these areas is far from trivial. On one hand, there are the areas that ontology developers determine to need extension, sometimes following end-users requests. These require not only extensive domain knowledge but also an increased effort to keep up with the novel knowledge being produced in that domain. On the other hand, there are the areas that could benefit from extension given their characteristics but have not been addressed by ontology developers yet. Discovering these areas in an automated fashion can be cast as a prediction problem, that identifies which ontology terms are more likely to be further extended based on previous extension events. By letting ontology developers and ontology learning systems focus on these areas, the present methodology can contribute to a faster rate of ontology evolution.

To support my research into automated change capturing I began with a preliminary investigation of ontology usage patterns. I then developed and evaluated two distinct approaches to predict ontology extension, one based on rules and heuristics, and another on supervised learning. Ultimately, the goal of this part of my research is not only to develop methods that are capable of accurately identifying the best possible ontology areas for extension, but also to determine the minimum set of ontology versions and other information needed to achieve this.

6.2 Related Work

Although there is a large body of work on ontology evolution (for a review see Leenheer & Mens (2008)), there are few works on the change capturing phase and to the best of my knowledge none on the prediction of ontology extension. Stojanovic *et al.* (2002) proposed an approach to ontology evolution that is based on optimizing the ontology

according to the end-users needs. They track end-users interactions with an ontology-based application to collect useful information that can be used to assess what the main interests of the end-users are. Their approach is then a usage-driven change discovery, which focuses on discovering anomalies in the design of an ontology, whose repairing improves its usability. They employ several measures, based on querying and browsing of an ontology-based application. Browsing-based measures are based on the user's browsing of links between ontology concepts. They define the usage of two concepts p and c as the number of times the link between them has been browsed, where c is a subconcept of a concept p. This concept is used in four measures for estimating the uniformity (balance) of the usage of a link regarding the link neighborhood: (1) SiblingUniformity represents the ratio between the usage of a link and the usage of all links, which have the common source node with that link (the so-called sibling links); (2) ChildrenUniformity stands for the ratio between the sum of the usage of all the links whose source node is the given node and the sum of the usage of a node through all incoming links into this node. (3) ParentUniformity is the ratio between the usage of a link and the usage of all links which have the common destination node with that link, and (4) UpDownUniformity characterizes the ratio between the usage of a link in two opposite directions, i.e. in browsing down and browsing up through a hierarchy. The classes with extreme values in these measures are then analyzed to determine the change they need.

Another usage-driven strategy was proposed by Haase *et al.* (2005) in the context of the evolution of multiple personal ontologies, which is based on a user's ratings of concepts and axioms, with the purpose of guiding the evolution process towards personalizing the ontologies.

Both of these approaches depend on the availability of detailed usage data or on a high degree of user involvement, which are not generally available for biomedical ontologies. However, for ontologies where an annotation corpus exists, we can consider these as usage measures.

Nevertheless, few biomedical ontologies possess large corpus of annotation, limiting the applicability of these approaches. Moreover, in the case of Stojanovic *et al.* (2002), the notion of uniformity although interesting from the point of view of tailoring the ontology to user's needs, is based on the notion that ontologies should have uniform structures. Although this may improve an ontology's usability in some domains, it is an incorrect assumption in the life sciences. Both our knowledge of the biomedical domain and the domain itself are anything but uniform.

Also relevant for this work is the investigation of ontology evolution in biomedical ontologies. In Ceusters (2009), the author applied a previously proposed strategy, Evolutionary Terminology Auditing (ETA) Ceusters & Smith (2006) to assess the quality of GO. These works follow a realist approach, where the extant ontology version is considered the benchmark against which a newer version is assessed. This strategy can be used not so much to demonstrate how good an individual version of a terminology is, but rather to measure how much it has been improved (or believed to have been improved) as compared to its predecessor. This is based on matches and mismatches between ontology versions, and their motivations, which are expressed by 17 possible configurations denoting the presence or absence of a term and whether the presence or absence of a term in a terminology is justified or unjustified. Of these 17 configurations only two correspond to a need for extension, in which an entity is missing and it is real and relevant for the ontology.
In another relevant work, Hartung *et al.* (2010) proposes an approach to automatically discover evolving or stable regions of ontologies. This approach is based on a cost model for changes between ontology versions and is able to identify regions that have been undergoing (or not) extensive changes. Although it is predictable that evolving regions will continue to evolve in the near future, if this kind of approach were to be used to derive candidates for extension, the stable regions would never again become targets for extension, regardless of their need for

extension, which makes this approach inappropriate as an exclusive means for identifying extension candidates.

6.3 Ontology usage patterns: a preliminary study

Before studying the evolution of the whole Gene Ontology, I performed a preliminary study focused on the general GOSlim. GOSlims are subsets of GO, that only include high-level terms and aim at summarizing GO. Each leaf term, the most specific terms in a GOSlim, is a representative of all its children terms and their annotations. This simple study assumed that the areas of GO that would benefit the most from automated refinement, would be the ones that are lagging behind in size, i.e. have less children terms, but still boast a significant annotation. These areas are called GO *hotspots*. To identify these areas this study calculated usage patterns, which are determined by the ratio between the frequency of annotation and the number of subclasses a GO term has.

To identify these *hotspots* within the 87 leaf terms of GOSlim generic, the ratio between the annotations made to a GOSlim term and the number of terms that it represents, i.e. the number of children it has, was calculated. This was done for five versions of GO spanning two years. Furthermore, to distinguish between manual annotations and computationally derived annotations, two different ratios for each version were calculated, one considering just the annotations that are made by curators, and another considering all annotations present in GOA.

For these two scenarios 17 distinct *hotspots* were identified (16 using all annotations and 4 using just manual ones) by applying a simple threshold whereby a GOSlim term is considered a *hotspot* if at any given time a 1.5 fold increase in the ratio of annotations per child was observed, that was not subsequently decreased. Figures 6.1 and 6.2

show the distributions of the annotation ratios for these terms in each scenario, for the five versions of GO.

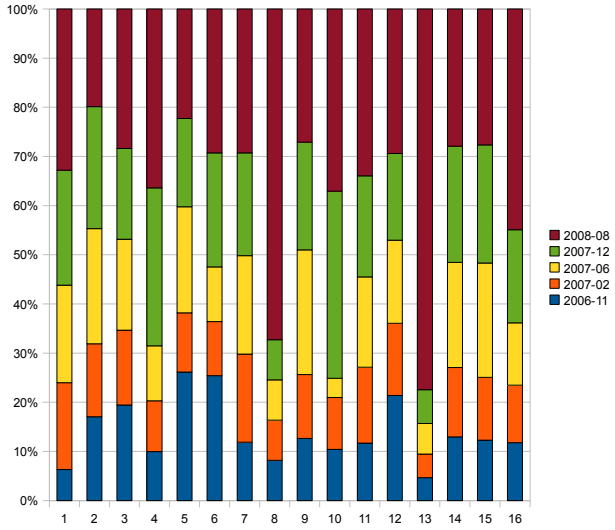

Figure 6.1: Distribution of the annotations per child ratio for the 16 *hotspots* found using all annotations.

1)reproduction; 2)generation of precursor metabolites and energy; 3) DNA metabolic process; 4)cell recognition; 5)cell death; 6)embryonic development; 7)cellular homeostasis; 8)cytoplasmic chromosome; 9)cell wall; 10)lipid particle; 11)cilium; 12)ion channel activity; 13)electron carrier activity; 14)antioxidant activity; 15)oxygen binding; 16)chaperone regulator activity

Each column corresponds to one *hotspot*, and each color corresponds to a version of GO. The size of each colored part of a column indicates the ratio of annotations vs. number of children found for the term in that ontology version. While some terms, like *cytoplasmic chromosome* and *ion channel activity* have only more recently passed the 1.5 fold increase threshold, other terms consistently maintain an elevated ratio for a period of 5 years, indicating that for those cases, increased usage for annotation did not result in a refinement of the ontology.
It is interesting to note that some of these *hotspots* match GO's Interest groups (e.g. embryonic development, viral reproduction, electron car-

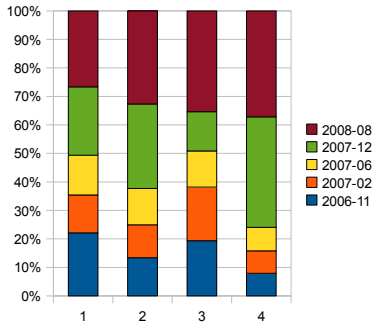

Figure 6.2: Distribution of the annotations per child ratio for the 4 *hotspots* found using manual annotations.

1) reproduction; 2)embryonic development; 3)viral reproduction; 4)lipid particle

rier activity, generation of precursor metabolites and energy). It is also noteworthy that the number of identified *hotspots* when using manual annotations is much lower than the number of *hotspots* identified when considering all annotations and that there is also considerable overlap between these two sets.

The low number of manually derived *hotspots* may be a reflection of a good articulation between GO development and GOA manual curation, which can mean that many GO terms are created when GOA curators need them for annotation purposes. On the other hand, when using all annotations, it was found that nearly 20% of the GOSlims leafs could benefit from enrichment.

6.4 Predicting Ontology Extension: a rule-based approach

Since many of the premises for good practice in ontology development can be applied for ontology evolution, Stojanovic & Motik (2002) proposed a series of guidelines for ontology evolution based on the guidelines for ontology development by Noy & Mcguinness (2000a). Two of

these are of particular interest for this work since they are concerned with ontology extension:

1. structure-driven: If a class has fewer children than its siblings, it may be a candidate for extension
2. data-driven: A class with many instances is a candidate for being split into subclasses and its instances distributed among newly generated classes.

Following the above mentioned guidelines I have devised a set of rules to apply to the prediction of the extension of GO. The rules aim at finding a partition of the set of classes that best separates classes that will be refined in a future version from those that will not. Since these guidelines are only concerned with refinement, this is the focus of the remainder of this analysis. Here, it was assumed that the latest ontology version is believed to be as correct and complete as possible Ceusters & Smith (2006). There are three types of rules, one structure-based and two data-based. The structure-based rules are derived from guideline 1:

Rule 1: A class with at most less x% subclasses than its siblings is a candidate for refinement

with x taking four evenly spaced values between 25 and 100%. The data-based rules are derived from guideline 2 but distinguish between the set of all annotations and the set of manually curated ones:

Rule 2: A class with at least x% more annotations than its siblings is a candidate for refinement

Rule 3: A class with at least x% more manual annotations than its siblings is a candidate for refinement

with the threshold x taking four evenly spaced values between 100 and 250%.

Distinguishing between these two sets of annotations is very relevant in the context of GO, since the set of manual annotations contains only

those that have been reviewed by a curator and can therefore be considered more reliable. Nevertheless, only about 3% of all annotations are manual which means they provide a narrower coverage.

6.4.1 Methods

The rules described above were applied to the twelve ontology versions used in 5.3.1. creating two sets of classes for each combination of rule and threshold, one corresponding to the classes above the threshold and the other to the ones below. Then to evaluate the predictive power of the rules, precision, recall and f-measure were computed for how well these sets reflect the real sets of refined and non-refined classes.

6.4.2 Results

Table 6.1 shows the results for predicting refinement in six months [1] for the thresholds x that generated the best results.

Although these results are overall poor, there is a marked difference between the performance of structure and data-based rules, with data-based rules having a higher precision for all three hierarchies and a higher recall in molecular function.

These rules were also applied to predicting the refinement for ontology branches as a whole, as opposed to the previous strategy that predicted refinement for individual classes. This follows from the observation that many of the new classes inserted in the biological process hierarchy are inserted as part of small subgraphs rather than single insertions. The focus was on the subgraphs that are rooted on classes at a depth of four due to the fact that most extension events occur at this depth or lower. However, the results obtained were comparable to those generated by predicting for individual classes.

[1] results for one and two years were similar - data not shown

Biological Process

Rule	Precision	Recall	F-measure
1 ($x = 75\%$)	$0.0772_{\pm 0.0317}$	$0.364_{\pm 0.0802}$	$0.127_{\pm 0.0479}$
2 ($x = 200\%$)	$0.220_{\pm 0.0185}$	$0.318_{\pm 0.0638}$	$0.256_{\pm 0.0128}$
3 ($x = 200\%$)	$0.242_{\pm 0.0302}$	$0.380_{\pm 0.0507}$	$0.292_{\pm 0.01714}$

Cellular Component

Rule	Precision	Recall	F-measure
1 ($x = 75\%$)	$0.0270_{\pm 0.0228}$	$0.381_{\pm 0.206}$	$0.0501_{\pm 0.0406}$
2 ($x = 200\%$)	$0.119_{\pm 0.109}$	$0.212_{\pm 0.246}$	$0.149_{\pm 0.148}$
3 ($x = 200\%$)	$0.199_{\pm 0.121}$	$0.374_{\pm 0.259}$	$0.252_{\pm 0.156}$

Molecular Function

Rule	Precision	Recall	F-measure
1 ($x = 75\%$)	$0.0122_{\pm 0.0033}$	$0.223_{\pm 0.0908}$	$0.0230_{\pm 0.0060}$
2 ($x = 200\%$)	$0.101_{\pm 0.0388}$	$0.406_{\pm 0.0357}$	$0.157_{\pm 0.0492}$
3 ($x = 200\%$)	$0.123_{\pm 0.0515}$	$0.526_{\pm 0.0573}$	$0.194_{\pm 0.0672}$

Table 6.1: Prediction results for the refinement of the Gene Ontology at 6 months. Shown values are averaged over all ontology versions, resulting from a total of 11 runs.

6.4.3 Discussion

These results emphasize that the current proposed guidelines for capturing change based on structure and data are not appropriate for handling a large and complex ontology such as the Gene Ontology. The guidelines represent an effort to ensure a balanced structure for the ontology, and given the size and evolving nature of the domain GO covers, its extension cannot be governed alone by these precepts. In fact, GO's Ontology Development group [1] has highlighted the processes used in the identification of areas that need to be developed:

- by working closely with the reference genome annotation group to ensure that areas that are known to undergo intense annotation in the near future are updated
- by listening to the biological community
- by ensuring that emerging genomes have the necessary classes to support their needs

If GO's change management regarding extension were to be made explicit, for instance as is the case for making a term obsolete where the reason is given, a more in-depth analysis could be performed and perhaps derive more accurate rules. Nevertheless, using the number of annotations rather than the number of subclasses yielded better results, which may be related to the fact that GO development is driven by need, which can be approximated by the rate of annotation, rather than by a process of homogeneization of structure. In fact, this difference was to be expected considering that in GO's domain the level of specificity of each branch is dependent on natural and scientific phenomena, which prevents the existence of an homogenous structure to the ontology. Such structure-based guidelines are however expected to function better in ontologies that follow a more conceptual approach.

In trying to predict ontology extension, particularly in the case of large biomedical ontologies, we are facing a multitude of variables, not

[1] http://wiki.geneontology.org/index.php/Ontology_Development_group_summary

only the advancement of biomedical knowledge and the current state of the ontology itself but also social and technical aspects. The extension of biomedical ontologies occurs via several different processes, and motivated by distinct needs, which cannot be apprehended by a 'one size fits all' rule. It becomes clear that to tackle this complexity, with numerous variables and their relations we need more sophisticated techniques.

6.5 Prediction of ontology extension: a supervised learning approach

My proposed methodology addresses change capturing by predicting ontology extension, and it was motivated by the fact that these changes can in principle be semi-automatically discovered by analyzing the ontology data and its usage. It is a supervised learning based strategy that predicts the areas of the ontology that will undergo extension in a future version, based on previous versions of the ontology. By pinpointing which areas of the ontology are more likely to undergo extension, this methodology can be integrated into ontology extension approaches, both manual and semi-automated, to provide a focus for extension efforts and thus contributing to ease the burden of keeping an ontology up-to-date.

6.5.1 Methods

6.5.1.1 Data

Fifteen versions of the Gene Ontology spanning a period of seven years were used. Table 6.2 identifies these versions, and describes a few general statistics about them. The versions have a six-month interval between them or as close to that as possible, since not all versions have a full database available from the Gene Ontology archive.

Table 6.2: Description of Gene Ontology versions

6pt4pt

ontology version	n. terms	n. relations	max depth	avg depth	deletions*	insertions*	total annotations	manual annotations
Jan 2005	17K	26K	17	6.8	N/A	N/A	6.0 M	0.50 M
Jul 2005	18K	28K	19	7.0	111	885	7.1 M	0.62 M
Jan 2006	19K	30K	18	7.0	42	1311	7.3 M	0.56 M
Jul 2006	20K	31K	18	7.0	20	578	9.0 M	0.56 M
Jan 2007	22K	35K	18	7.2	97	2079	10.4 M	0.62 M
Jun 2007	23K	38K	18	6.9	131	1454	12.4 M	0.66 M
Jan 2008	24K	40K	18	4.9	153	1674	19.0 M	0.73 M
Jul 2008	25K	44K	18	4.9	104	807	23.0 M	0.78 M
Jan 2009	27K	47K	18	4.9	17	1415	24.7 M	0.79 M
Aug 2009	28K	51K	18	5.0	77	1487	33.0 M	0.87 M
Jan 2010	29K	54K	19	4.9	61	1476	33.5 M	0.91 M
Jul 2010	32K	57K	15	3.9	31	1302	60.5 M	1.06 M
Jan 2011	33K	60K	15	4.01	106	2698	54.4 M	1.23 M
Jul 2011	34K	63K	15	4.03	48	1208	63.8 M	1.35 M
Jan 2012	36K	65K	15	4.05	32	1113	77.8 M	1.41 M

*with respect to the version in the line above

6.5.1.2 Extension Prediction Strategy

The intuition behind the proposed strategy is that information encoded in the ontology or its annotation resources can be used to support the prediction of ontology areas that will be extended in a future version. This notion is inspired by change capturing strategies that are based on implicit requirements. However in the existing change capturing approaches, these requirements are manually defined based on expert knowledge. This method attempts to go beyond this, by trying to learn these requirements based on previous extension events using supervised learning.

In the test case using GO, the attributes used for learning are a series of ontology features based on structural, annotation or citation data. These are calculated for each GO term and then used to train a model able to capture whether a term would be extended in a following version of GO.

Structural features give information on the position of a term and the surrounding structure of the ontology, such as height (i.e. distance to a leaf term), number of sibling or children terms. A term is consid-

Table 6.3: Features and feature sets used for supervised learning

Type	Feature	all	simple structure	uniformity	annotations	direct	indirect	*bestA*	*bestB*
		Feature set							
Structural	*dirChildren:* descendants* of a term at a distance of one	+	+			+			
	allChildren: all descendants* of a term	+	+				+	+	+
	height: maximum distance to a descendant	+	+						
	siblings: number of terms that share at least one parent*	+	+						
Annotation	*dirManAnnots:* direct annotations given a manual evidence code	+			+	+			
	dirAnnots: direct annotations	+			+	+			
	allManAnnots: annotations (direct and inherited) given a manual evidence code	+			+		+	+	+
	allAnnots: annotations (direct and inherited)	+			+		+	+	+
Citation	*PubMed:* number of articles in PubMed mentioning the term or its children six months before ontology version	+						+	+
Hybrid	*ratioAll:* ratio between *allAnnots* and *allChildren*	+							+
	ratioDir: ratio between dirAnnots and dirChildren	+							
	siblingsUniformity: ratio between *allAnnots* for the term and the sum of *allAnnots* for its siblings	+		+					
	parentsUniformity: ratio between *allAnnots* for the term and the sum of *allAnnots* for its parents	+		+					
	childrenUniformity: ratio between *allAnnots* for the term and the sum of *allAnnots* for its children	+		+					

* in the *is_a* and *part_of* hierarchies

ered to be direct child if it is connected to its parent by an *is_a* or *part_of* relation, but the total of children of a term encompasses all descendants regardless of the number of links between them. Annotation features are based on the number of annotations a term has, according to distinct views (direct vs indirect, manual vs all). Direct annotations are annotations made specifically to the term, whereas indirect annotations are annotations made to a parent of the term, and thus inherited by the term. Manual annotations correspond to those made with evidence codes that reflect a manual intervention in the evidence

supporting the annotation, while the full set of annotations also includes electronic annotations. Citation features are based on citation of ontology terms based on external resources, in our case PubMed. Finally hybrid features combine some of the previous features into one single value. These features can be mapped onto the change discovery types: structural features belong to their homonymous change discovery type; annotations features can be seen as both data and usage based, since they can be interpreted as both ontology instances and ontology usage; and citation features correspond to the discovery-driven change, since they are derived from external sources. In total 14 features were defined and then grouped into five sets (see Table 6.3): *all*, *structure*, *annotations*, *uniformity*, *direct*, *indirect*, *best*A and *best*B. The first three sets are self-explanatory. Uniformity set features were based on Stojanovic (2004), with annotations representing usage. The *direct* set joins direct features of terms, in terms of children and annotations, whereas the *indirect* set joins the same kind of features in their indirect versions. The *best* sets were based on the best features found after running the prediction algorithm for individual features.

Due to the complexity of ontology extension, a framework for the outlining of ontology extension in an applicational scenario was established. This framework defines the following parameters:

- Extension type:
  - **refinement**, where a term is considered to be extended if it has novel children terms
  - **enrichment**, where a term is considered to be extended if it has novel hierarchical relations to existing terms
  - **extension**, where a term is considered to be extended if it has novel children terms and/or novel hierarchical relations to existing terms
- Extension mode:
  - **direct**, where a term is considered to be extended if it has new children terms (according to extension type)

  - **indirect**, where a term is considered to be extended if it has any new descendant terms (according to extension type)
- Term set:
  - **all** terms
  - terms at a given **depth** (maximum distance to root)
  - terms at a given distance to **GOSlim** terms
- Time parameters:
  - **nVer**, the number of versions used to calculate the features
  - **ΔFC**, the time interval(in number of ontology versions) between versions used to calculate features and version used to verify extension (i.e. in our dataset, a ΔFC of two equals a time interval of one year, since the chosen ontologies are spaced by six months.)

By clearly describing the ontology extension process according to this framework, it becomes possible to accurately circumscribe the ontology extension prediction efforts.

The datasets used for classification were then composed of vectors of attributes followed by a boolean class value, that corresponded to extension in the version to be predicted, according to the used parameters. To compose the datasets we need not only the parameters but also an initial set of ontology versions to be used to calculate features and the ontology version to calculate the extension outcome (i.e. class labels). So given a set of sequential ontology versions $O_v = \{O_1, ..., O_n\}$, we need to choose one ontology version to predict extension, O_e, and then based on time parameters $nVer$ and ΔFC, select the set of ontologies to be used to calculate features. For example, for a set of ontologies $O_v = \{O_1, ..., O_6\}$, if we chose O_6 to predict extension, along with $nVer = 3$ and ΔFC=2, the set of ontologies to calculate features will be $O_f = \{O_2, O_3, O_4\}$.

Several supervised learning algorithms were tested, namely Decision Tables, Naive Bayes, SVM, Neural Networks and Bayesian Networks,

using their WEKA implementations Hall *et al.* (2009). For Support Vector Machines, the LibSVM implementation was used with an RBF kernel and optimized the cost and gamma parameters through a coarse grid search. For Neural Networks, the chosen implementation was the Multilayer Perceptron, with the number of hidden layers equal to $(attributes + classes)/2$, a training time of 500 epochs, and a coarse grid search to optimize the learning rate was performed. Regarding Bayesian Networks, the probabilities were estimated directly from the data, and different search algorithms were tested, namely Simulated Annealing, K2, and Hill Climbing. Furthermore, taking into consideration that there are many more terms that are not extended than terms that are, between two sequential ontology versions, which creates unbalanced training sets, the SMOTE algorithm was used Chawla *et al.* (2002). SMOTE (synthetic minority over-sampling technique), is a technique that handles unbalanced datasets by over-sampling the minority class and under-sampling the majority class that has been shown to support better classification results for the minority class.

6.5.1.3 Evaluation

A simple approach was used to evaluate the ontology extension prediction strategy : compare the predictions to the actual extension of the Gene Ontology in a future version. To this end another time parameter was needed:

- $\Delta\mathbf{TT}$, time interval between versions used for training and testing

This time parameter is used to create the test set, by shifting the ontology versions according to ΔTT. So for instance, given a set of ontologies $O_v = \{O_1, ..., O_5\}$ and using $nVer = \Delta$FC=ΔTT= 1, the training and test sets would correspond to the those in Figure **??**. Although there may be an overlap in the ontology versions used in a particular training/testing setup, the ontology versions used to determine the class values are always distinct, ensuring that our setup in

unbiased.

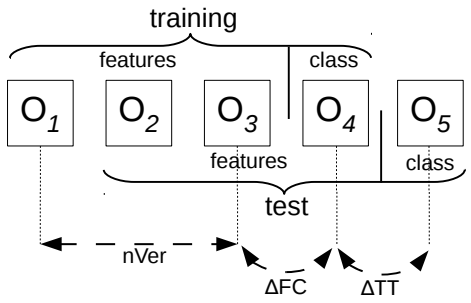

Figure 6.3: Example of ontology versions to use for training and testing with $nVer = 3$, ΔFC= 1 and ΔTT= 1.

This approach allows us to compare the set of proposed extensions to real ones that actually took place in a future version of the ontology. Precision, recall and f-measure metrics can be calculated by using the real extension events observed in the more recent ontology version as our test case. These metrics are based on the number of true positives, false positives, true negatives and false negatives. A true positive is an ontology class that our supervised learning strategy identified as a target for extension, and that was indeed extended in the test set, whereas a false positive although having also been identified as a target for extension, was not actually extended. Likewise, a false negative is an ontology class which was not identified as a target for extension, but was in fact extended in reality, whereas a true negative was neither identified as a target nor was it extended in the test set. Precision corresponds to the fraction of classes identified as extension targets that have actually been extended, while recall is the fraction of classes identified as extension targets out of all real extensions. F-measure is a measure of a test's accuracy that considers both precision and recall.

6.5.2 Results

When trying to predict ontology extension the focus should not only be on which features are best predictors, but also on how to design the learning process to best support the prediction. Consequently, the goal is not only trying to find the best prediction setup in terms of features and machine learning algorithms, but also in terms of our strategy's parameters.

6.5.2.1 Parameter optimization

A first step in the experiments was to determine the best term set to use, and to investigate if this was influenced by different parameters. To this end, the following term sets within each GO ontology were tested: all terms, all terms with a depth of 3, 4 and 5, all GO Slim general terms, all GO Slim general leaf terms, all terms at a depth of 1 from the GO Slim general leaf terms, under the same sets of parameters (see Table 6.4).

Table 6.4: Average term set sizes

	Average term set size		
Term Sets	Biological Process	Cellular Component	Molecular Function
all	15928	2272.8	8265.6
depth=3	97.07	21	154
depth=4	374	112.47	495.33
depth=5	849	178.47	1093.67
GOSlim	65.27	31.67	-
GOSlim leaves	54.07	26.07	-
GOSlim leaves depth=1	1189.93	758.73	-

To provide a simple basis for this first analysis the focus was shifted to just the biological process hierarchy, a single feature *allChildren*

and WEKA's *Decision Table* algorithm with attribute selection using *BestFirst*. Results are presented (unless otherwise specified) using the average f-measure obtained using all possible setups derived from the 15 GO versions available, since a large number of combination of different parameters is being analyzed. So for instance, when using $nVer = 1$, ΔFC= 2 and ΔTT= 2, we get a total of ten runs for our prediction evaluation, whereas using $nVer$=3, ΔFC= 1 and ΔTT= 1 we get only six runs.

Before comparing term sets, the trends between parameter sets need to be analyzed. First, in what concerns extension types and modes (see Table 6.5) it is clear that indirect extension is predicted with much more success (0.49-0.86) than direct extension (0.1-0.27). Furthermore, in regards to comparing refinement to enrichment and generic extension, enrichment is poorly predicted, with a performance around 0.20-0.30. The performance for indirect refinement and extension in term sets derived from depth performance is comparable (0.63-0.78), whereas in GO Slim sets refinement is better predicted (0.65-0.86 vs. 0.62-0.65).

Table 6.5: Comparison of extension types and modes

Term Sets	refinement direct ($\mathcal{A}$)	refinement indirect ($\mathcal{B}$)	enrichment indirect ($\mathcal{C}$)	extension indirect ($\mathcal{D}$)
all	0.0999 ± 0.07817	0.4919 ± 0.03250	0.2009 ± 0.09838	0.4674 ± 0.03577
depth=3	0.2704 ± 0.22514	0.7896 ± 0.05400	0.2955 ± 0.22057	0.7495 ± 0.05059
depth=4	0.2176 ± 0.17606	0.7083 ± 0.03660	0.3429 ± 0.17947	0.6790 ± 0.04012
depth=5	0.2313 ± 0.14730	0.6348 ± 0.04879	0.2898 ± 0.14780	0.6268 ± 0.05476
GOSlim	0.2024 ± 0.22988	0.8637 ± 0.05889	0.1722 ± 0.21296	0.6530 ± 0.30708
GOSlim leaves	0.1635 ± 0.21344	0.8553 ± 0.06710	0.1003 ± 0.17292	0.6470 ± 0.30122
GOSlim leaves depth=1	0.1523 ± 0.13830	0.6529 ± 0.06636	0.3168 ± 0.10540	0.6243 ± 0.07201

Values are average and standard deviation f-measure for all runs using the 15 ontology versions and a Decision Table algorithm, in the biological process hierarchy. Time parameters: $nVer = 1$, ΔFC= 1, ΔTT= 1

To clarify this difference, the average extended proportion for each extension type was calculated (see Table 6.6 for the values for the term set at depth=4), i.e. the average proportion of extended terms for all GO versions. This showed that the proportion of extended terms is higher for biological process, independently of extension type, followed by cellular component and molecular function, and that the proportion of refined terms is higher than enriched terms, independently of GO term type. This can have an impact on training since there are fewer examples of enrichment.

Table 6.6: Average extended proportion for Gene Ontology according to extension type

	refinement	enrichment	extension
biological process	0.293	0.103	0.292
cellular component	0.122	0.027	0.124
molecular function	0.076	0.013	0.077

Values are averaged for all GO term at depth=4 for the 15 ontology versions with an indirect extension mode.

Regarding the time parameters (see Table 6.7) and using indirect extension and refinement, the differences are less marked. An increase in the number of versions ($nVer$) used to calculate the feature values from one to three does not significantly alter the results, and when the interval between versions for feature extraction and extension is extended it results in an increase in overall performance of about 0.02-0.06.

In general, when comparing term sets considering the best sets of parameters ($\mathcal{B}$,$\mathcal{C}$ and $\mathcal{E}$, see Tables 6.5 and 6.7), it is clear that smaller term sets show a better overall performance. For the remainder of the analysis the focus will be on two term sets, *depth=4* and *GO Slim leaves depth=1*, which will be referred to as *depth* and *GOSlim* respectively. These sets were chosen to cover both term set strategies and

Table 6.7: Comparison of time parameters

Term Sets	$nVer$=1, ΔFC= 1, ΔTT= 1 ($\mathcal{B}$)	$nVer$=1, ΔFC= 2, ΔTT= 2 ($\mathcal{E}$)	$nVer$=3, ΔFC= 1, ΔTT= 1 ($\mathcal{F}$)	$nVer$=3, ΔFC= 2, ΔTT= 2 ($\mathcal{G}$)
all	0.4919 ± 0.03250	0.5301 ± 0.01627	0.4890 ± 0.03550	0.5301 ± 0.01627
depth=3	0.7896 ± 0.05400	0.8177 ± 0.05422	0.8152 ± 0.04293	0.8005 ± 0.07808
depth=4	0.7083 ± 0.03660	0.7520 ± 0.03340	0.7267 ± 0.04551	0.7437 ± 0.04113
depth=5	0.6348 ± 0.04879	0.6962 ± 0.04093	0.6526 ± 0.04885	0.7101 ± 0.03863
GOSlim	0.8637 ± 0.05889	0.9020 ± 0.07523	0.8264 ± 0.06208	0.8869 ± 0.08646
GOSlim leaves	0.8553 ± 0.06710	0.9004 ± 0.06908	0.8378 ± 0.05228	0.9046 ± 0.07896
GOSlim leaves depth=1	0.6529 ± 0.06636	0.6748 ± 0.07166	0.6624 ± 0.06722	0.7021 ± 0.04651

Values are average and standard deviation f-measure for all runs using the 15 ontology versions and a Decision Table algorithm, in the biological process hierarchy. Extension mode: refinement, indirect

provide a reasonable size set without sacrificing too much performance. Also from now on, all results will pertain to refinement and indirect extension, since they represent the primary goal of finding areas of the ontology to extend. Considering time parameters the best overall performers (setup $\mathcal{G}$: $nVer$=3, ΔFC= 2, ΔtTT= 2) were used.

6.5.2.2 Features

The next step in the experiment was to compare different features and feature sets. Table 6.8 presents the average and standard deviation f-measure values for all features and feature sets using our standard setup.

When using single features, the best performers are *allChildren*, *height* and *allManAnnots*, with average f-measure values around 0.74 in the *depth* set and 0.69 in the *GOSlim* set. When using sets of features, in the *depth* set the top performers are *indirect*, *bestB* and *all*, with values between 0.75 and 0.76, whereas in *GOSlim* they are *all*, *bestB* and *bestA*, with values between 0.77 and 0.78. Using feature sets insetad of single features has a positive impact on performance in the

GOSlim set, which is not noticeable in the *depth* set.

Table 6.8: Feature and Feature Sets performance for Biological Process

		Term set	
	Features	depth	*GOSlim*
Single	dirChildren	0.6723 ± 0.02641	0.6662 ± 0.04143
	allChildren	0.7437 ± 0.04113	0.7021 ± 0.04651
	height	0.7426 ± 0.03482	0.6854 ± 0.04387
	sibsUniformity	0.5814 ± 0.15741	0.5283 ± 0.15760
	parentsUniformity	0.6336 ± 0.03964	0.5430 ± 0.17153
	childrenUniformity	0.6469 ± 0.05440	0.5899 ± 0.08983
	dirAnnots	0.4857 ± 0.15008	0.4964 ± 0.06482
	dirManAnnots	0.4838 ± 0.10863	0.4748 ± 0.05278
	allAnnots	0.7335 ± 0.03663	0.6821 ± 0.03579
	allManAnnots	0.7452 ± 0.02882	0.6965 ± 0.04940
	PubMed	0.5960 ± 0.03933	0.6552 ± 0.04709
	ratioAll	0.6850 ± 0.04231	0.6192 ± 0.03266
	ratioDir	0.5735 ± 0.11476	0.5856 ± 0.03939
Sets	all	0.7459 ± 0.03675	0.7801 ± 0.0525
	structure	0.7431 ± 0.02543	0.6906 ± 0.04546
	uniformity	0.6523 ± 0.06109	0.5727 ± 0.19389
	annotations	0.7396 ± 0.02893	0.6949 ± 0.04771
	direct	0.6661 ± 0.03684	0.6569 ± 0.05436
	indirect	0.7641 ± 0.03242	0.6883 ± 0.06412
	bestA	0.7415 ± 0.04270	0.7704 ± 0.04450
	bestB	0.7550 ± 0.03049	0.7750 ± 0.04265

Values are average and standard deviation f-measure for all runs using the 15 ontology versions and a Decision Table algorithm. Time parameters: $nVer = 3$, ΔFC= 2, ΔTT= 2.

6.5.2.3 Gene Ontologies

So far, predicting refinement has only been expounded for the biological process ontology. Tables 6.9 and 6.10 summarize the results obtained for the molecular function and cellular component hierarchies, showing the top three features and feature sets for each term set. For molecular function only results for the term set based on *depth* are given since there is no *GOSlim* subset.

Although average f-measure is generally lower for both molecular function and cellular component, than for biological process, *allChildren*

and *allManAnnots* continue to be among the best features. Furthermore, for cellular component the *GOSlim* set shows a worse overall performance than the *depth* set, in disagreement with what happens in biological process.

Table 6.9: Summary of Feature and Feature Sets performance for Cellular Component

		Term set	
	Features	depth	*GOSlim*
Single	allManAnnots	0.7085 ± 0.07487	0.6068 ± 0.06908
	allChildren	0.6800 ± 0.11041	0.5650 ± 0.09469
	ratioAll	0.6604 ± 0.04485	0.4636 ± 0.02932
	height	0.6450 ± 0.08744	0.5248 ± 0.08186
Sets	bestB	0.7210 ± 0.08485	0.5174 ± 0.08370
	bestA	0.7155 ± 0.09198	0.4758 ± 0.11213
	annotations	0.7046 ± 0.08523	0.6198 ± 0.03661
	all	0.6916 ± 0.11118	0.4367 ± 0.14839
	structure	0.6890 ± 0.13975	0.5985 ± 0.04716

Values are average and standard deviation f-measure for all runs using the 15 ontology versions and a Decision Table algorithm. Time parameters: $nVer = 3$, ΔFC= 2, ΔTT= 2.

Table 6.10: Summary of Feature and Feature Sets performance for Molecular Function

		Term set
	Features	depth
Single	allChildren	0.6650 ± 0.07957
	allManAnnots	0.5898 ± 0.07267
	height	0.5633 ± 0.08577
	dirChildren	0.5577 ± 0.06710
	allAnnots	0.5572 ± 0.08084
Sets	bestA	0.6441 ± 0.04625
	indirect	0.6395 ± 0.07485
	bestB	0.6285 ± 0.06971
	all	0.6218 ± 0.04873
	structure	0.6168 ± 0.06450

Values are average and standard deviation f-measure for all runs using the 15 ontology versions and a Decision Table algorithm. Time parameters: $nVer = 3$, ΔFC= 2, ΔTT= 2.

6.5.2.4 Supervised Learning Algorithms

In addition to Decision Tables, chosen due to their simplicity, several other commonly used supervised learning algorithms, namely Naive Bayes, SVM, Neural Networks (Multilayer Perceptron) and Bayesian Networks, were also tested using their WEKA implementations. Figure 6.4 shows a plot for precision and recall for the best feature sets using these algorithms.

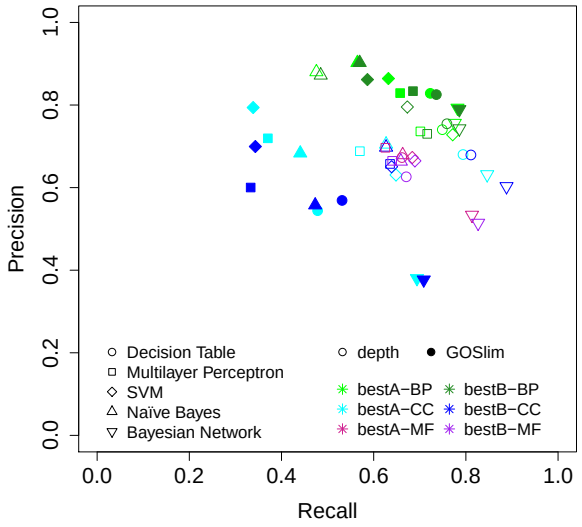

Figure 6.4: Average precision and recall for several supervised learning algorithms using the *bestA* and *bestB* feature sets, depth and GO Slim based term sets and $nVer$=3, ΔFC= 2, ΔTT= 2 in all three GO hierarchies.

When applying different learning algorithms, overall biological process still has the best performance, followed by molecular function and cellular component. Likewise, the general performance in the *GOSlim* term set is better than the one in the depth term set for biological process, whereas it is the reverse for cellular component.
Looking in with more detail at the biological process results, the difference between feature sets is small, so we will not distinguish between

them in our analysis. Naive Bayes gives the top precision values (0.87-0.90) but the lowest recall (0.48-0.57), whereas Bayesian Networks have the highest recall (0.78-0.79) with precision values between 0.74 and 0.79, which correspond to average f-measures between 0.76 and 0.79. SVM, Decision Tables and Multilayer Perceptron have performances in between these with both recall and precision values clustered around 0.70.

In molecular function, the highest precision is given by Multilayer Perceptron at 0.70 for *bestA*, and Multilayer Perceptron, SVM and Naive Bayes for *bestB* at 0.66-0.67. The highest recall is found in *bestB* by Bayesian Networks at 0.83. Best average f-measure is achieved by SVM at 0.66 for both *bestA* and *bestB*.

In cellular component, there is a marked difference between the performance in the depth term set and in the *GOSlim* set, with the latter having in general a much lower recall, around 0.40, except when using Bayesian Networks, where recall rises to around 0.7, but at the cost of precision. There is also a visible difference between term sets, with *bestB* having in general a lower precision for the *GOSlim* set, which is not apparent in the depth term set. In the depth term set the best performing algorithms are Decision Tables and Bayesian Networks, with recall around 0.8 and precision above 0.6. Decision Tables achieves the top performance with an average f-measure of 0.72 for *bestA*.

6.5.2.5 Comparative evaluation

To provide a basis for comparison, Stojanovic's browsing uniformity measures Stojanovic (2004) were implemented and evaluated on predicting ontology evolution for GO. Annotation frequency was used as a proxy for link usage. Since this strategy does not identify targets for extension, but rather ranks classes according to their uniformity, the evaluation was performed by plotting precision-recall curves for all ontology versions used. Fig 6.5 shows precision/recall plots for children

uniformity, using one version of the ontology to calculate uniformity and predicting refinement for a following version in the dataset, alongside the plots for the proposed prediction strategy best configuration ($\mathcal{G}$, $bestB$). For both cases the term set based on a depth of 4 was used along with a distance between training and testing of two versions.

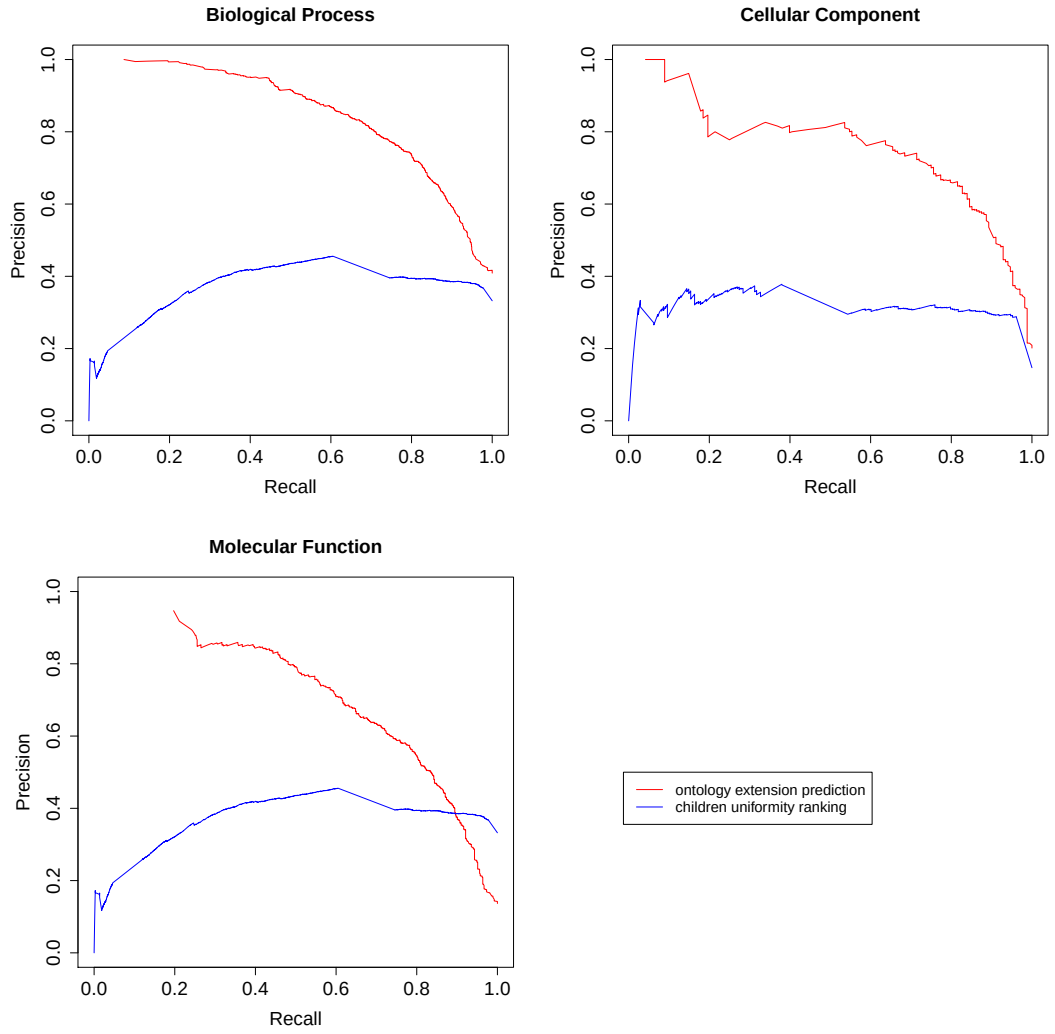

Figure 6.5: Precision/Recall plots for refinement prediction based on Stojanovic's children uniformity and our own strategy.

For plotting the proposed strategy, instead of relying on the binary

labels output by the classifier, the probabilities for each instance to be true (i.e. refined) were used. This was done so that the generated plots are more directly comparable to those produced by the uniformity strategy, allowing a more granular calculation of precision at different recalls supporting a threshold based evaluation. Consequently, the presentation of the results of our strategy in these plots differs from the presentation in previous tables.

The prediction results for all ontologies were combined together in the plotting of the Precision/Recall plots to provide a better visualization of results. As the plots clearly show, the proposed strategy has a considerably improved performance in all three GO ontologies, with curves closer to the top right corner, which are indicative of both higher precision and recall. The uniformity strategy performed worse in all cases, except at higher recall values in molecular function.

The other uniformity strategies (parents and siblings) have an even lower performance than that of children uniformity.

6.5.3 Discussion

Change capturing through prediction of ontology extension is a complex issue, due to the inherently complex nature of ontology extension itself. Ontology extension can be motivated by implicit or explicit requirements, which have very different mechanisms. Implicit requirements are in principle easier to predict since they do not change between ontology versions, whereas explicit requirements, which are created by experts to adapt the ontology to a novel conceptualization or change in the domain, are much harder to predict. The strategy proposed in this thesis, by virtue of being based on learning using past extension events, cannot distinguish between these two types, and thus attempts to predict extension regardless of it being motivated by implicit or explicit requirements. To capture both kinds of requirements the proposed strategy uses a set of ontology based features that not

only contemplates intrinsic features, such as structural ones, but also extrinsic ones, such as annotations and citations.

The assumption that extension can be predicted based on existing knowledge, either in the form of the ontology itself or its usage, is acceptable regarding the more common extension events, but is not applicable to extension events that are the result of deep restructuring or revision of existing knowledge. These extension events are part of a complex ontology change that also includes deletions and modifications. As such, these more complex changes are not the object of this strategy. In fact, one of the proposed strategy's goals is to speed up the process of accomplishing the simpler extensions, to give experts more time and resources to focus on the more complex events.

One very relevant aspect of the evaluation strategy used here is that the results are compared to the real extension events that occurred in more recent versions of the ontology. This means that although some predictions are conceptually correct, they may not have yet been included in the ontology version used for testing and will thus be considered incorrect. This will have an impact on precision values, since the strategy might be capturing needed but still unperformed extensions, and then be considering them to be incorrect in our evaluation. Due to this line of thought, we might then give preference to strategies that increase recall even if at the cost of precision. However, this could have the negative effect of including many incorrect predictions in our output, which is not desirable in a semi-automated ontology extension system. As such the evaluation was based on f-measure, to provide balanced precision and recall.

A basic requirement of the presented strategy is to be able to access several versions of the ontology to consider. The minimum set of ontology versions it requires is two: one which will be used to calculate the features, and a second one, more recent than the first, from which the class labels will her extracted to train the model. It then becomes crucial to define the interval between the versions to use. In this test

case using the Gene Ontology, versions with an interval of at least six months were used based on the intuition that a smaller interval would not provide sufficient extension examples to be able to train a model. This intuition was shown to be a good approximation, since as seen in Table 6.11, when using monthly versions there are in fact a very low number of positive examples.

6.5.3.1 Parameters

Due to the complexity of ontology extension, particularly in such a large ontology as the GO, an extension prediction strategy has to account for several parameters that help circumscribe our effort. One such parameter, extension type, was designed to capture the different types of extension: refinement and enrichment. It was found that refinement is considerably easier to predict than enrichment, with refinement having a greater average f-measure by between 0.3 and 0.7. There are two likely explanations for this difference: on one hand, there are many more refinement events between ontology versions than there are enrichment events (see Table 6.6), which will provide a better support for supervised learning; on the other, the features used may be better correlated to refinement than to enrichment.

Another parameter related to extension, is its mode, direct or indirect. Predicting direct extension, i.e. exactly which terms will be extended in a future version, should be the ultimate goal of an ontology extension prediction strategy. However this was proven to be a difficult task, which is unsurprising given the multitude of different processes that can lead to extension, and also the fact that on average new terms correspond to about 5% of all terms in an ontology version (see Table 6.2). This follows the trend found in our previous work Pesquita & Couto (2011), where we analyzed the extension of GO and found that insertions of new terms often occur together.

To address this issue the prediction efforts were focused on slices of the ontology, and extension that happens within the subgraphs rooted in

terms within these slices was defined as indirect extension. Focusing only on the term sets thus defined greatly improved the performance (Table 6.5), with average f-measures for the prediction of refinement of biological process increasing from 0.49 to 0.65-0.86 depending on the term set considered.

Predicting for a subset of the ontology is supported by the previous finding Pesquita & Couto (2011) that extension frequently happens by branches and that introducing terms closer to the root has a large impact on the overall structure of the ontology. Consequently, determining which term sets to use must be a compromise between enough specificity to be useful, but enough generality to provide a good enough balance of positive and negative examples. Six such subsets were determined, following two distinct approaches: based on distance to root and based on GO Slim general.

Distance to root was chosen for its simplicity in creating a middle layer of GO terms. However, since terms at the same distance to the root do not always have the same degree of specificity, GO Slim general was also used as a basis for our other strategy. The purpose of using GO Slim general was to capture a similar degree of specificity among terms, detailed enough to provide a useful prediction and general enough to allow for branch extension prediction. Three different sets were tested within each approach, each yielding different term set sizes. Since molecular function does not have a GO Slim general, we only tested distance to root (*depth*) based sets.

For both approaches, the smaller the dataset the better the results. This can be due to the fact that in smaller data sets there is a better balance of positive and negative instances, which despite the use of SMOTE to balance the training sets, still has an impact on training the models. However, very small term sets are not of interest, since they would not provide enough specificity to change capturing for ontology extension. Considering this the focus was on the term set defined by terms at a distance of one from GO Slim leaf terms, which corresponds to an average term set size of 1189 for biological process and 758 for

cellular component, and on the term set defined by terms at a distance of four to the root, which corresponds to sizes around 370, 460 and 100, for biological process, molecular function and cellular component respectively.

The final parameters in our strategy are those related with time: $nVer$, ΔFC and ΔTT. The influence of the number of versions used to derive the features was found to be minimal. Regarding the intervals between versions for feature and class, and for training and testing, increasing those intervals from six months to one year resulted in an increase in performance (about 0.03 to 0.06), which is likely due to the fact that the number of positive examples is larger when considering a larger interval between versions. Considering these results, the setup of $nVer$=3, ΔFC= 2 and ΔTT= 2 was chosen.

6.5.3.2 Features

Although the parameters previously discussed represent the basis of our strategy, by defining exactly on what the prediction is focusing, it is the features used to support prediction that are essential to be able to capture extension events. Using the best parameter setup a set of thirteen single features, also arranged into eight sets, was investigated. In the *depth* term set, the single features *allChildren* and $allManAnnots$ were among the top performers for the three GO hierarchies. But in the *GOSlim* for biological process feature sets performed better than single features, whereas in cellular component this difference was not apparent. However, the feature sets composed of the best single features ($bestA$ and $bestB$) were shown to provide the better performances across the board, with the exception of the *GOSlim* set in cellular component. It is interesting to note that although using just structural or just annotation based features can provide in most cases a performance comparable to combining them, which can sim-

plify our strategy, using a combination of the best single features can in some cases improve performance.

One of the most obvious patterns obtained from these results is that terms with a lot of children terms or a lot of total annotations tend to be extended. It is arguable that for larger subgraphs, the probability of an extension event occurring is greater, given that there are more terms in it. However, to support the theory that the only factor involved is indeed the number of terms in the subgraph (i.e allChildren), we would have to consider that the probability of extension for any given term is equal. Intuitively, this does not appear to be a valid assumption, since it would mean that the extension of GO does not follow any particular direction. Nevertheless, this possibility was investigated by comparing the distribution of real refinement events for $allChildren$ intervals, with the probability density function of a binomial distribution for at least one success for the same $allChildren$ intervals. Figure 6.6 shows that the two distributions are significantly different, thus supporting the notion that although the number of children has an influence in the refinement probability, the probability of refinement is not the same for all terms. From these results we can hypothesize that the number of children a term has is related to its probability of refinement, because it reflects an increased interest in that area of the ontology.

Furthermore, the total number of annotations is influenced by the total number of children, since the annotations of the children contribute to the total number of annotations of the parent. To take this into account, the feature ratioAll was created to mitigate the influence of the number of children on the annotation data. Although this resulted in a decrease in f-measure of around 6%, compared to either feature separately, it is still a better performance than most other features. This gives further support to the notion that areas which attract a larger

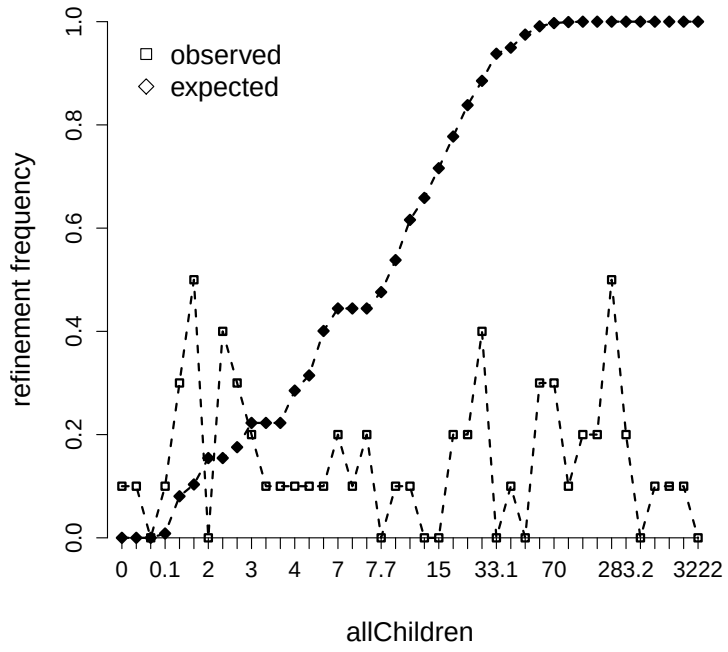

Figure 6.6: Relation between number of *allChildren* and refinement probability.

The label 'observed' corresponds to the real observed refinement events, whereas 'expected' to the refinement proportion expectable following the binomial distribution. Presented values correspond to the GO version of June 2010, but other versions present a very similar behavior. *allChildren* and refinement values are averaged within intervals of size 10. These intervals were calculated by ordering the terms according to the their *allChildren* number in ascending order, and then generating equal sized intervals.

interest (in this case patent in the number of annotations) become the object of more refinement events.

Although these simple notions appear quite intuitive, and a simple generic rule based on the number of children could in principle be derived, in order to support automated change capturing the best separation possible between targets and non-targets for refinement needs to be established, which is best achieved by employing supervised learning.

6.5.3.3 Supervised Learning

The results discussed so far were all based in Decision Tables, a simple supervised learning algorithm. Other algorithms, such as SVM, Neural Networks and Bayesian Networks, were also tested and although they were capable of providing a better performance, and specifically in the case of SVM and Neural Networks of being parametrized to privilege either precision or recall, Decision Tables was still able to provide generally good results comparatively, without requiring parameter optimization.

The performance of Bayesian Networks was of special interest, since the attributes used are not independent, but in fact are temporally related when we consider multiple ontology version for feature extraction. For instance the value of *allChildren* in one version depends on its value in the previous one. However, no marked difference between Bayesian Networks and other approaches was found, so this dependency appears to not be very relevant for our current strategy.

Another particularly interesting aspect is that most machine learning algorithms, including the ones that were used, assume that instances are all independent and identically distributed. However, the dataset instances correspond to GO terms which are hierarchically related through the GO structure. Although the inclusion of features that describe the neighboring area tried to capture this aspect (e.g. siblings, and all the uniformity features), we still believe it was not properly contemplated by the proposed setup. The hierarchical relations between instances may be affecting the experiments considering the full set of terms, since they are not being captured by the representation. In the subset of terms dataset, their influence would not be as strong, since there are fewer hierarchical relations between instances.

6.5.3.4 Comparative Evaluation

To complete this evaluation the proposed strategy was compared to the one proposed byStojanovic *et al.* (2002) based on uniformity. In general, the uniformity based strategy performed worse than the one proposed here. This however is a consequence of Stojanovic's approach having been designed to support the manual extension of an ontology that adapts to user's needs, whereas in this setting the ontology models knowledge about a domain whose extension is caused by many different aspects. Curiously, when transforming the uniformity metrics into features for classification, a better performance is achieved (Table 6.8) than when using them as intended by the authors, as a simple criteria for ranking.

Applying extension prediction

The output of the extension prediction methodology is a list of ontology classes, which are the roots of subgraphs that correspond to ontology areas which have been predicted as good candidates for extension. This methodology is applicable to the most simple yet most frequent type of ontology change, the addition of new elements. It is not suited to predict more complex changes such as a reorganization of an entire branch of the ontology. As such, the ontology extension prediction can be used to speed up the process of extension in these simpler cases, by allowing ontology developers and/or ontology learning systems to focus on smaller areas of the domain. This frees the experts to spend more time focusing on the more complex changes that cannot be predicted.

Automated ontology learning systems can also use the list to focus their efforts on the identified areas. For instance, most ontology learning systems employ a corpus of scientific texts as input, and their performance is tightly coupled to the quality of such corpora. If the candidate list is used to guide the creation of specific corpora for the

areas to extend, it can have a positive impact on the performance of such strategies.

We have chosen to highlight three examples of the results given by the ontology extension prediction system, two successful ones (Fig. 6.7 and Fig. 6.8), where the predicted areas were in fact extended in the version for which extension was predicted, and one indirectly successful one (Fig. 6.9), where although the extension did not occur when predicted, it did in fact happen at later versions of the ontology.

In Fig. 6.7, extension was predicted for the subgraph rooted in "macromolecular complex assembly". Since it is indirect extension that is being predicted, the addition of new subclasses can occur at any point in the subgraph. In this case, the GO term has four direct subclasses, and all of them gained new subclasses in the future version for which we were predicting. In Fig. 6.8, extension was predicted for the area of "cell pole". In the version used to train the model, "cell pole" had two subclasses "apical pole of neuron" and "basal pole of neuron" but in the version for which extension was predicted, "cell pole" gained a whole new branch rooted on a new subclass for "cell tip". These two examples showcase two different extension patterns: in the first, extension occurs throughout the subgraph, whereas in the second it corresponds to the addition of a single but large branch.

In Fig. 6.9 extension was predicted in the subgraph of "lipid transporter activity" for the version of January 2010, but no extension took place. However in later versions of July 2010 and January 2011, extension did occur by the addition of two new sub-subclasses. This is an example of how this evaluation strategy may be too stringent when considering these cases false positives, since they can eventually undergo extension at later versions.

6.5.3.5 Consecutive version prediction

Although six months was used as the minimum interval between consecutive ontology versions, a small study using monthly versions was also conducted. As a first step the average number of refinements made in consecutive monthly versions between May 2010 and October 2010 (six versions) within each GO ontology (Table 6.11) was investigated.

Table 6.11: Average number of refined and non-refined GO terms

GO ontology	refined terms	non-refined terms
depth=4	consecutive versions	
biological process	84.00 ± 22.967	594.5 ± 18.554
molecular function	9.50 ± 9.604	449.25 ± 259.479
cellular component	3.25 ± 3.418	114.5 ± 66.130
GOSlim leafs depth=1	consecutive versions	
biological process	118.25 ± 41.583	1156.75 ± 41.583
cellular component	6.5 ± 7.762	604.0 ± 348.785
depth=4	six month separated versions	
biological process	105.89 ± 22.605	224.22 ± 36.0917
molecular function	41.89 ± 16.010	466.89 ± 20.572
cellular component	11.78 ± 2.779	92.56 ± 44.919

The number of refined terms between consecutive versions is much lower than using a six month interval, as expected. In the molecular function and cellular component ontologies, this has a great impact on prediction, since between some versions there were actually no refinements at all. This of course precludes prediction for these cases, but even when there are refinements, the number of positive training examples is still much lower than the negatives, making it either impossible to train a model, or making the model over fitted, and hence have a lower performance on the prediction task.

6.5.3.6 Evolution of prediction

All presented results have been averages for all predictions made using a given setup. However it is also interesting to verify if there is any

trend in extension prediction, so individual f-measure values for all three GO ontologies were plotted using the standard setup (Decision Tables, $bestA$, $nVer = 3$, ΔFC= 2, ΔTT= 2, refinement, indirect). Figure 6.10 shows this plot, where it can be observed that for biological process, there is very little variation across time, whereas for molecular function and cellular component there is greater variation.
It is then possible to hypothesize that this could be due to variations in the number of positive examples between different versions, which could be impacting the training of the model. To investigate this the percentage of positive examples within each dataset was calculated and plotted in Figure 6.11.

6.6 Conclusions

In this chapter I have presented an investigation on the automation of the first step of ontology evolution, the change capturing phase. I tested two approaches: a rule based one and a supervised learning one. Both approaches are based on predicting areas of the ontology that will undergo extension in a future version, by applying either rules or learning over features of previous ontology versions.

The rule based approach was derived from guidelines for ontology development proposed in the literature. The efficacy of three different rules was tested on the prediction of the extension of the Gene Ontology, one based on the number of subclasses of GO terms and two others based on the number of annotations (all or just manual). All rules proved to be unsuccessful, supporting the conclusion that the extension of GO is too complex to be captured by such simple rules. In face of this complexity, the learning approach was expected to provide better results, and this was in fact the case with average f-measure

reaching 0.79 for prediction of refinement for a subset of relevant biological process GO terms. The supervised learning approach was tested using a broad selection of parameters and features, which served as an investigation of the minimum set of versions and features needed to provide useful prediction results. Although the best results were obtained using a set of structural and annotation features from multiple ontology versions, good results were also obtained using a single ontology version and simple structural features. This is crucial to the applicability of the proposed strategy to other biomedical ontologies, since most lack such a rich annotation corpus as GO's and are not updated with the same frequency.

I find that two particular characteristics of the proposed strategy can be improved, namely the selection of ontology versions to use and the selection of the term set. Both of these can benefit from recent works on ontology evolution Hartung *et al.* (2009, 2010) from which we can gather useful information to guide the selection process. For the ontology versions, as discussed above, there is a need for a minimum of changes between versions to allow for the training, and by using these works we can pinpoint ontology versions that have enough changes between them. In what concerns the term set, we can benefit from the identification of stable and evolving regions of the ontology, and thus dynamically define distance to root based on this criteria, i.e. for stable regions we predict for terms further away from the root, whereas for evolving regions we stay closer to the root.

Another interesting avenue to improve this strategy is concerned with the machine learning algorithms. Most of these algorithms, including the ones that were used, assume that instances are all independent and identically distributed. However, the dataset instances correspond to GO terms which are related hierarchically through the GO structure. The hierarchical relations between instances may be affecting the experiments considering the full set of terms, since they are not being

captured by the representation. In the subset of terms dataset, their influence would not be as strong, since there are few hierarchical relations between instances. Furthermore, when using more than one ontology version for features, the same features for each version are temporally related. These challenges can be addressed by employing a more complex strategy based on (Sharan & Neville, 2008), which accounts both for relations between instances, and temporal variation of attributes by combining a Relational Bayesian classifier (RBC) with a Relational probability tree (RPT).

To the best of my knowledge, there has been no previous research into the prediction of ontology evolution and its use for change capturing. Integrating the proposed strategy into a semi-automatic ontology engineering framework can bring numerous advantages to ontology engineers and developers, particularly in large ontologies, which are very common in biomedicine. Using the candidate classes that are returned by this method we can build focused corpora. This on one hand reduces the amount of data ontology learning methods need to process, increasing the speed of the whole process which can be of particular relevance in domains where the speed of ontology evolution plays a crucial role, such as epidemiology. On the other, it narrows down the domain of each corpus to be analyzed, which can have a positive impact on the efficacy of ontology learning systems. One issue of these systems is that they generate many spurious new candidate classes since they are based on term recognition methods that are unable to differentiate between generic terms and domain terms. By providing a more focused input corpus to begin with, this issue can be partially avoided.

Nevertheless, human experts will always be required, especially to handle the more complex modeling tasks, but clever integration of the proposed strategy with ontology learning methods is expected to dramatically decrease the workload of ontology engineers and drive the

cost of ontology extension in the biomedical domain down, both in terms of time and resources.

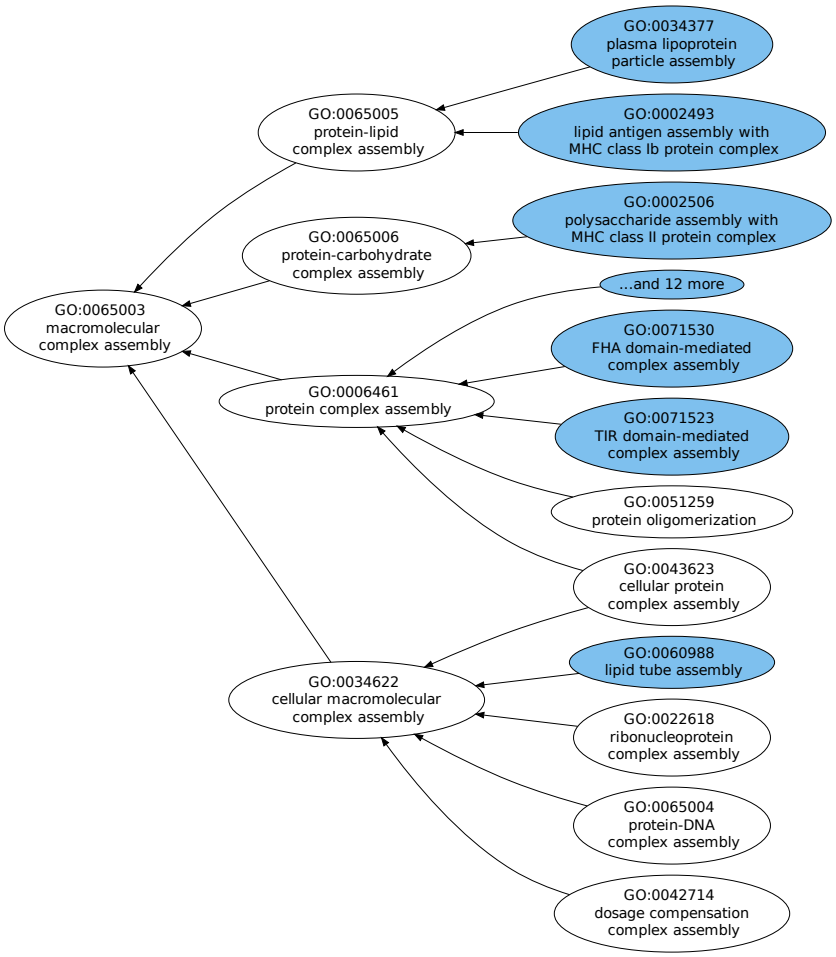

Figure 6.7: Example of predicted extension in the Molecular Function hierarchy.

Extension was predicted for the root term and occurred at a distance of two edges, in every subclass.

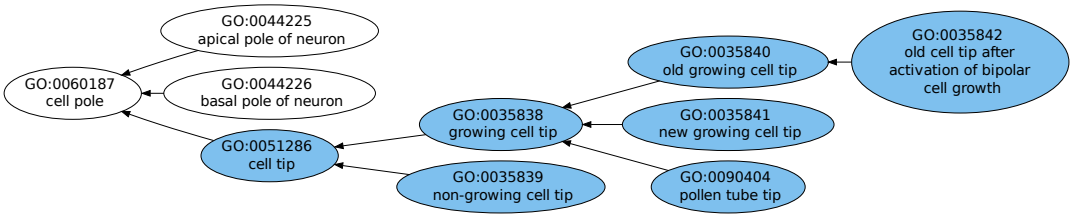

Figure 6.8: Example of predicted extension in the Cellular Component hierarchy.

Extension was predicted for the root term and occurred at a distance of one edge, with the addition of a whole new branch.

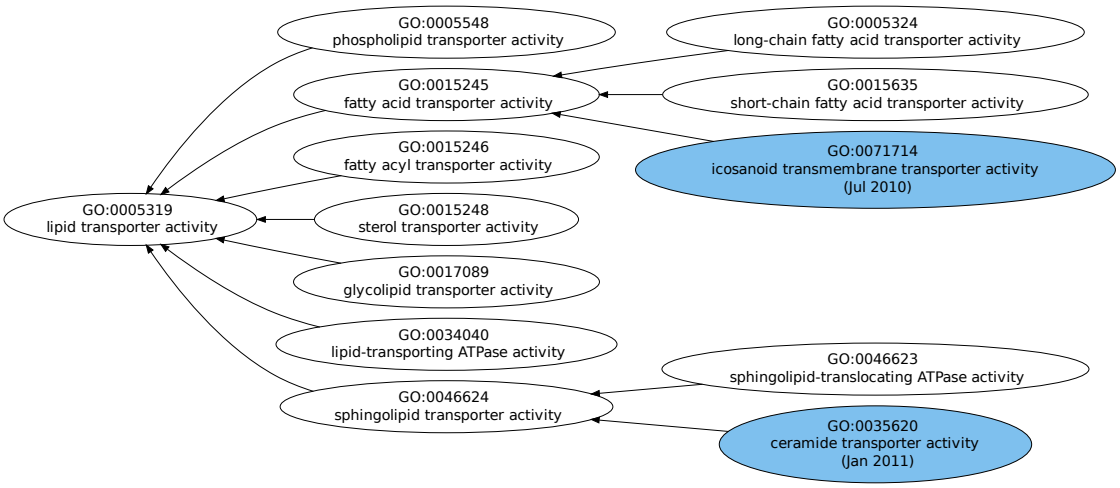

Figure 6.9: Example of predicted extension in the Biological Process hierarchy.

Extension was predicted for the root term and although it did not occur in the version for which it was predicted (January 2010), it did in fact occur in later versions, with the addition of one new sub-subclass in July 2010 and another in January 2011.

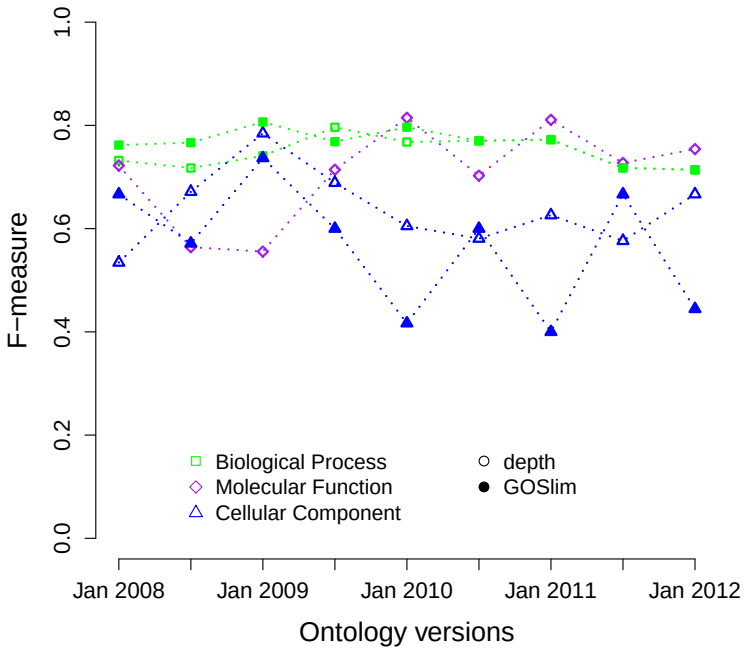

Figure 6.10: F-measure for refinement prediction for separate ontology versions using Decision Tables with the $bestA$ feature set and $nVer = 3$, ΔFC= 2, ΔTT= 2.

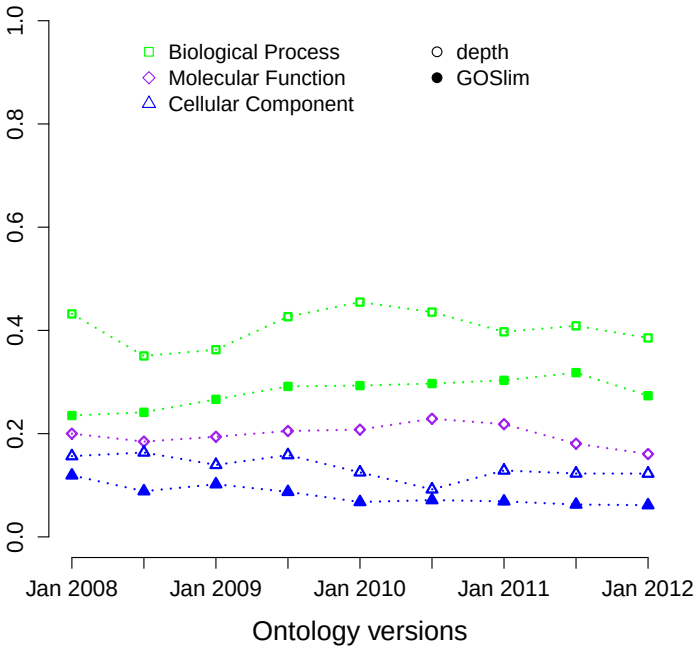

Figure 6.11: Percentage of positive examples for training models for refinement prediction for separate ontology versions.

Chapter 7

Exploiting ontology vocabulary in ontology learning and enrichment

Ontology learning from text can be used as an integral part of an ontology extension approach, provided that the learned ontologies are integrated into the one to extend. One of the key tasks of ontology learning for extension is the identification of novel candidate classes. Retrieving these classes from text is often based on automated term recognition (ATR) strategies that are able to process a text corpora to find the relevant concepts they contain. ATR strategies usually function by first identifying all terms within the texts and then ranking them according to their relevance to the domain. Term relevance measures are used to rank terms according to their relevance to the domain. These measures can be classified into two types (Kageura & Umino, 1996): *unithood* measures and *termhood* measures.
Unithood is "the degree of strength or stability of syntagmatic combinations or collocations," and termhood is "the degree to which a linguistic unit is related to (or more straightforwardly, represents) domain-specific concepts." Unithood measures include mutual information, log likelihood, t-test, and the notion of 'modifiability' and its

variants (Deane, 2005; Wermter & Hahn, 2005). A few studies, such as (Deane, 2005), leverage on n-gram sequences to identify potential terms.

Termhood measures are frequency-based and use reference corpora. They include information retrieval methods such as TF-IDF (Robertson & Jones, 1976); domain relevance (Navigli & Velardi, 2004) which compares the term frequency in the corpus with its frequency in an external corpora from distinct domains; domain pertinence (Sclano & Velardi, 2007) and domain consensus (Navigli & Velardi, 2004).

Recent studies have focused on hybrid approaches, that combine unithood and termhood into a single score. These include C-vale/NC-value (Frantzi *et al.*, 1996), Termextractor (Sclano & Velardi, 2007) and many others. Two recent comparative evaluation studies of ATR methods have been conducted (Zhang *et al.*, 2008) (Korkontzelos *et al.*, 2008). Both concluded that the performance of the methods varied with the corpus used and that approaches including termhood measures consistently performed better.

In the context of ontology extension, guaranteeing that the identified terms belong to the domain in question is crucial. However, existing termhood measures are tailored to use reference corpora instead of ontologies. To address this issue I developed a term relevance measure FLOR that can be used to calculate the 'domainness' of a term by measuring its relatedness to ontology terms. Interestingly, the approach of FLOR can also be used to enrich an ontology with relations between existing terms by calculating their relatedness.

Another strategy to filter recognized terms is to remove from the candidate term lists terms that have been identified by named entity recognition methods as belonging to a distinct domain. To test this hypothesis I developed a method based on a popular named entity recognition system for biomedical entities, the GENIA tagger (Tsuruoka *et al.*, 2005).

7.1 Related Work

FLOR is inspired by FiGO, a system for the the identification of Gene Ontology (GO) terms in PubMed abstracts developed by Couto *et al.* (2005) for BioCreAtIvE (Blaschke *et al.*, 2005). FiGO is based on the frequency of the words present in the GO vocabulary. The more frequent a word is, the smaller is its relevance to the final score of the GO term. This strategy is derived from term relevance measures used in term identification approaches.
The GENIA tagger is a named entity recognition system specifically tuned for biomedical text. It analyzes English sentences and outputs the base forms, part-of-speech tags, chunk tags, and named entity tags. The GENIA tagger is capable of identifying the following biomedical entities: protein, DNA, RNA, cell line and cell type.

7.2 FLOR: A term relevance measure

FLOR calculates the relevance of a term in an ontology domain by calculating the relatedness between the term and ontology concepts. It relies on textual information contained in ontologies and uses two information theory notions: the evidence content of a word, and the information content of an ontology concept.

7.2.1 Components of ontology textual information

An ontology vocabulary is the set of all textual information contained in an ontology in the form of labels, synonyms and definitions.
Ontology concepts usually have several textual descriptors (e.g., name, synonyms, definitions). So an ontology concept description can be stated as follows:

$$D(c) = \{d_0, ..., d_n\} \tag{7.1}$$

where d_n are its textual descriptors: name, synonyms and definition.
A concept descriptor is a unit of text related to an ontology concept ,

i.e. its label, synonym or definition.

7.2.2 Evidence content of a word

The evidence content of a word is calculated as the negative logarithm of the relative frequency of a word in the ontology vocabulary.

$$EC(w) = -log f(w) \tag{7.2}$$

where $f(w)$ is the frequency of the word in the vocabulary of an ontology. In FiGO, Couto et al. calculate the frequency of a word based on its occurrence in GO names and synonyms. FLOR considers all textual descriptions available, so it also uses definitions when available. Like FiGO, it also filters out common English words, but uses a more elaborate syntactic processing before the computation of word frequencies. The TreeTagger package (Schmid, 1995) is used to tokenise, lemmatise and perform part-of-speech tagging on the text retrieved from GO term's names, synonyms and definitions. TreeTagger is not prepared to handle complex biochemical names, so these remain unlemmatised and are tagged as nouns. Each noun token is then further tokenized, using special characters and delimiters, with the specific intent of handling biochemical entities and other complex words. For instance, the term *dibenzo-p-diazine* generates three entries: *dibenzo-p-diazine*, *dibenzo* and *diazine*.
Tokens are then reduced to their stem, using the Snowball algorithm, an updated version of the Porter Stemmer algorithm (Porter, 2001). Furthermore, since stemming algorithms are unable to handle cases such as *pancreas* and *pancreatic*, FLOR uses a simple rule whereby words are merged to their longest common prefix, when their length is greater than 5 characters, and their longest common prefix can only be at most 3 characters shorter than the longest word it represents. This results in the aggregation of words that fall under the same stem into

a single entry, reducing the bias towards uncommon forms of common root words. For instance *interactively* and *interaction* are both merged into the entry *interact*.
The final frequency of a word corresponds to the number of terms that contain it in their descriptors. This means that a word that appears multiple times in the name, definition or synonyms of a term is only counted once, preventing bias towards terms that have many synonyms with similar word sets.

7.2.3 Information content of an ontology concept

The information content (IC) of a concept c (Resnik, 1998) is a measure of how likely the concept is to occur in a given corpus, which can be quantified as the negative log likelihood,

$$IC(c) = -logp(c) \tag{7.3}$$

where $p(c)$ is the frequency of occurrence of c in a specific corpus. The information content can be given as a number bounded between 0 and 1 by using the relative information content (Pesquita *et al.*, 2008a) as follows:

$$IC(c) = -\frac{logf(c)}{logN} \tag{7.4}$$

where N is the size of the corpus.

For ontologies where no corpus is available, we can calculate the IC of a concept using the number of its descendants, $IC_{children}$, so that concepts with few descendants are considered more informative and vice-versa.

7.2.4 Calculating the relatedness between two ontology concepts

The relatedness between two ontology concepts given by FLOR corresponds to the maximum similarity between all possible combinations of descriptors:

$$FLOR(c_1, c_2) = max_{d1 \in D(c_1), d2 \in D(c_2)} Sim(d_1, d_2) \tag{7.5}$$

where c_1 and c_2 are two ontology concepts with $D(c_1)$ and $D(c_2)$ as their sets of descriptors. As is shown below, the relatedness between two concepts is not symmetrical, so c_1 is considered to be the query term, and c_2 to be the target term.

7.2.4.1 Similarity between text descriptors

The similarity between two descriptors of ontology concepts, $Sim(d_1, d_2)$, can be given by equation 7.6.

$$Sim(d_1, d_2) = \{. \begin{array}{ll} if(d_1 \supset d_2), & Sim_{EM} \\ else, & Sim_{PM} \end{array} \tag{7.6}$$

If d_2 is found to be contained in d_1 as a substring, we consider it an exact match, if not we consider it a partial match. In case of an exact match, the final score is given by a function of the information content of c_2, weighted by a factor α:

$$Sim_{EM} = (1 - \alpha) + (\alpha \ IC(c_2)) \tag{7.7}$$

Since Sim_{EM} returns values bounded between 0 and 1, α reflects the portion of the score that is influenced by the concept IC. This implies that the similarity between two descriptors is not symmetrical, since the substring relation is one-way only. In case of a partial match, the final score is given by a weighted Jaccard similarity, $SimPM$, which is calculated by transforming each descriptor into a vector of words, each

word weighted by its evidence content, and then dividing the sum of the EC of the words both descriptors share, by the sum of the EC of all words contained in both descriptors.

$$Sim_{PM} = \frac{\sum_{w \in (d_1 \cap d_2)} EC(w)}{\sum_{w \in (d_1 \cup d_2)} EC(w)} \tag{7.8}$$

where w are the words contained in the descriptors d_1 and d_2, and EC is the evidence content of a word. It provides a measure of the relevance of the words shared by both descriptors versus the total relevance of their words.

7.2.5 Calculating the relatedness between a textual term and ontology concepts

To calculate the relatedness between an ontology concept and a textual term a very similar approach is used, where an artificial concept is created with a single descriptor which corresponds to the textual term. Then the approach for the relatedness between concepts is used.

7.3 FLOR in ontology learning

FLOR can be used to provide a term relevance measure to automated recognition methods for ontology learning. The score returned by FLOR can give a measure of how relevant it is for the ontology. Another strategy to ensure that the identified terms belong to the domain in question is to filter out easily recognizable named entities that belong to other domains. The strategy proposed here would begin by first extracting noun-phrases that can potentially be candidate terms, then removing known named entities from this list and finally ranking the remaining terms using FLOR.
Biomedical named entity recognition of genes has already achieved a

high performance with the first ranked system in BioCreative 2 reaching 87.21% f-measure (Wilbur *et al.*, 2007). However genes are just one kind of biomedical entity and depending on the domain of the ontology to extend, recognizing other kinds of entities can be helpful. For instance, texts within the domain of GO will likely reference several kinds of biomedical entities including genes, proteins, RNA, cells, laboratory procedures, etc. As a proof of concept for this strategy I implemented a term recognition system based on the coupling of the GENIA tagger and FLOR for the domain of GO.

7.3.1 Methods

Abstracts retrieved from PubMed based on the output of the Extension Prediction module are parsed by the GENIA tagger. The output of GENIA is then processed to filter out irrelevant terms. First, a list of all noun phrases is compiled from the POS tagger output. Then, all noun phrases that exactly correspond to named entities recognized by GENIA are removed. The remaining noun-phrases are run through FLOR and ranked according to their relatedness to GO concepts.

7.3.2 Test case

The abstract in Figure 7.1 was retrieved from PubMed by the query 'neuron differentiation', taking this concept as an area in need of extension. It was run through GENIA resulting in the POS tagging in Figure 7.2 and the named entity recognition in Figure 7.3. Then all noun phrases that were not identified as named entities were ranked with FLOR to generate candidate concepts. Table 7.1 shows the resulting scores for the candidates.

In this case, all top candidates are within the scope of GO, with some corresponding to existing GO terms, for instance 'neuronal morphogenesis' is covered by 'cell morphogenesis involved in neuron differentiation' and 'axonal outgrowth' by 'axon extension'. However all other

J Cell Biol. 2011 May 16;193(4):769-84. Epub 2011 May 9.

Sarm1, a negative regulator of innate immunity, interacts with syndecan-2 and regulates neuronal morphology.

Chen CY, Lin CW, Chang CY, Jiang ST, Hsueh YP.

Institute of Molecular Biology and 2 Molecular and Cell Biology Program, Taiwan International Graduate Program, Graduate Institute of Life Sciences, National Defense Medical Center and Institute of Molecular Biology, Academia Sinica, Taipei 115, Taiwan.

Abstract

Dendritic arborization is a critical neuronal differentiation process. Here, we demonstrate that syndecan-2 (Sdc2), a synaptic heparan sulfate proteoglycan that triggers dendritic filopodia and spine formation, regulates dendritic arborization in cultured hippocampal neurons. This process is controlled by sterile α and TIR motif-containing 1 protein (Sarm1), a negative regulator of Toll-like receptor 3 (TLR3) in innate immunity signaling. We show that Sarm1 interacts with and receives signal from Sdc2 and controls dendritic arborization through the MKK4-JNK pathway. In Sarm1 knockdown mice, dendritic arbors of neurons were less complex than those of wild-type littermates. In addition to acting downstream of Sdc2, Sarm1 is expressed earlier than Sdc2, which suggests that it has multiple roles in neuronal morphogenesis. Specifically, it is required for proper initiation and elongation of dendrites, axonal outgrowth, and neuronal polarization. These functions likely involve Sarm1-mediated regulation of microtubule stability, as Sarm1 influenced tubulin acetylation. This study thus reveals the molecular mechanism underlying the action of Sarm1 in neuronal morphogenesis.

Figure 7.1: Abstract retrieved for the query 'neuron differentiation'

candidates are viable novel GO terms. Although this is just a a test case, it illustrates how FLOR can be used as a term relevance measure in ATR strategies for ontology extension.

7.4 FLOR in ontology enrichment

This section presents an evaluation of FLOR on its ability to retrieve related concepts within the same ontology. The strategy described below can be applied to ontology enrichment, by uncovering relations between ontology concepts.

[NP Dendritic arborization] [VP is] [NP a critical neuronal differentiation process] . [ADVP Here] , [NP we] [VP demonstrate] [SBAR that] [NP syndecan-2] ([NP Sdc2]) , [NP a synaptic heparan sulfate proteoglycan] [NP that] [VP triggers] [NP dendritic filopodia and spine formation] , [VP regulates] [NP dendritic arborization] [PP in] [NP cultured hippocampal neurons] . [NP This process] [VP is controlled] [PP by] [NP sterile & # 945] ; and [NP TIR motif-containing 1 protein] ([NP Sarm1]) , [NP a negative regulator] [PP of] [NP Toll-like receptor 3] ([NP TLR3]) [PP in] [NP innate immunity signaling] . [NP We] [VP show] [SBAR that] [NP Sarm1] [VP interacts] [PP with] and [VP receives] [NP signal] [PP from] [NP Sdc2] and [VP controls] [NP dendritic arborization] [PP through] [NP the MKK4-JNK pathway] . [PP In] [NP Sarm1 knockdown mice] , [NP dendritic arbors] [PP of] [NP neurons] [VP were] [ADJP less complex] [PP than] [NP those] [PP of] [NP wild-type littermates] . [PP In] [NP addition] [PP to] [VP acting] [ADVP downstream] [PP of] [NP Sdc2] , [NP Sarm1] [VP is expressed] [ADVP earlier] [PP than] [NP Sdc2] , [NP which] [VP suggests] [SBAR that] [NP it] [VP has] [NP multiple roles] [PP in] [NP neuronal morphogenesis] . [ADVP Specifically] , [NP it] [VP is required] [PP for] [NP proper initiation and elongation] [PP of] [NP dendrites] , [NP axonal outgrowth] , and [NP neuronal polarization] . [NP These functions] [VP likely involve] [NP Sarm1-mediated regulation] [PP of] [NP microtubule stability] , [SBAR as] [NP Sarm1] [VP influenced] [NP tubulin acetylation] . [NP This study] [ADVP thus] [VP reveals] [NP the molecular mechanism] [VP underlying] [NP the action] [PP of] [NP Sarm1] [PP in] [NP neuronal morphogenesis] .

Figure 7.2: POS tagging by GENIA

7.4.1 Methods

FLOR can be adapted to retrieve concepts that are related to an input concept. Given that bio-ontologies are generally large, a preparatory step is needed to derive a set of target concepts for which this method will be applied. The set of target concepts is the union between all:

1. concepts whose name or synonym is contained in the query concept descriptors - exact match concepts, and
2. the top k related concepts to the input concept, that are not in-

Table 7.1: FLOR scores for top candidate concepts after named entity removal

Candidate concepts	FLOR score
microtubule stability	0.46
dendritic arborization	0.45
neuronal morphogenesis	0.45
axonal outgrowth	0.45
neuronal polarization	0.44
dendritic spine formation	0.4
tubulin acetylation	0.38
dendritic filopodia formation	0.31

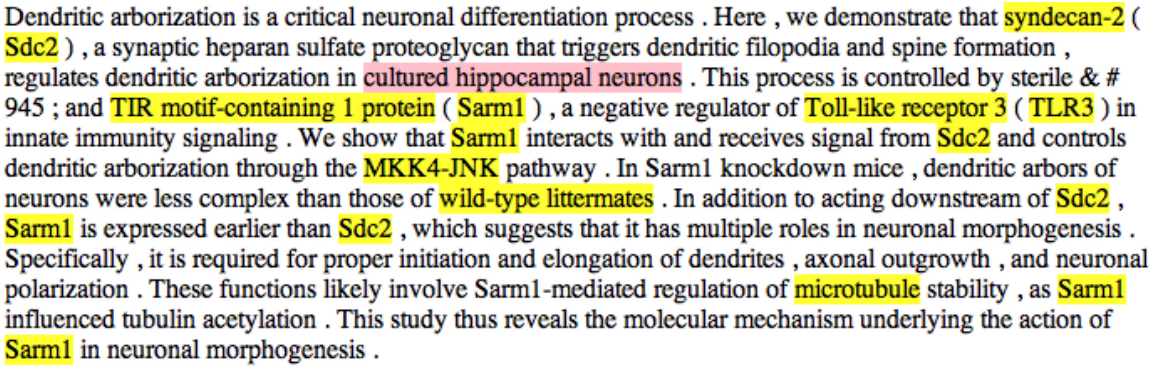

Figure 7.3: NER by GENIA

cluded in the previous set - partial match concepts.

These top k concepts correspond to concepts ranked by the highest sum of EC from the words that they share with the input concept. The candidate target concepts include the exact match concepts and the partial match concepts that share relevant words with the query concept. The relatedness score is calculated for each pair (query concept - target concept), and the target concepts are ranked according to the relatedness score.

7.4.2 Evaluation

The Gene Ontology (March 2010 release) was used to evaluate the performance of FLOR in two tasks: (1) calculating the relatedness between two GO terms. (2) retrieving GO terms related to a query GO term. Although GO is usually described as three separate ontologies, there is an increasing number of relations across them, which enables the use of the three ontologies as a single one, for the purpose of testing ontology enrichment.

7.4.2.1 Dataset

The Gene Ontology is particularly suited to the evaluation of FLOR since GO crossproducts provide a set of terms that are related (Mungall

Crossproducts dataset	Term pairs
bp_x_mf	12
mf_x_mf	27
mf_x_cc	28
bp_x_cc	245
cc_x_cc	284
bp_x_bp	736
Total	1332

Table 7.2: Number of term pairs derived from each GO crossproducts dataset.

et al., 2010). GO crossproducts are definitions of GO terms by composition using two other terms. The intuition behind using GO crossproducts to evaluate the extraction of relations between GO terms is that if a given term t_1, can be defined by composition of two other terms, t_2 and t_3, then t_1 can be said to be related to the terms t_2 and t_3, since it refers them. Consider the term *mitotic spindle elongation* , which is defined as a crossproduct in the *obo* format, as follows:

> [Term] id: GO:0000022 ! mitotic spindle elongation
> intersection_of: GO:0051231 ! spindle elongation
> intersection_of: part_of GO:0000278 ! mitotic cell cycle

This triplet generates two related query term-target term pairs: 'mitotic spindle elongation - spindle elongation ' and 'mitotic spindle elongation' - 'mitotic cell cycle '.
Two kinds of crossproducts datasets were used: the intra-hierarchy, or self, datasets, where a term of a given hierarchy is defined by composition of two terms of the same hierarchy, and the inter-hierarchy datasets, where a term of a given hierarchy is defined using one term from the same hierarchy and one term from a distinct hierarchy. A total of 6 datasets were generated from these cross-products sets, with a total of 1332 pairs of terms. Table 7.2 presents each dataset size.

The corpus of annotations provided by GOA can also be used to calculate information content. The probability of a term occurring in the GOA corpus is given by its frequency of annotation. In this evaluation

we use all annotations, regardless of their evidence code, to compute the information content. Also, we considered α the weighting factor of Sim_{EM} to be 0.2. The intuition behind this, is to allow for relatively high scores when an exact match is found, since the minimum score allowed for an exact match is 0.8.

7.4.2.2 Systems used for comparison

FLOR was compared to two previously proposed methods:

1. FiGO, a method proposed by Couto *et al.* (2005) to uncover GO terms in text;
2. substring strict matching, a method used by Ogren *et al.* (2004) to find compositional relations between GO terms.

FiGO calculates the confidence level for a GO term occurring in a text by the dividing the local evidence content (the sum of the evidence content of the words shared by the term and the text) by the term's evidence content.
The strict matching algorithm simply checks if the the name or synonyms of the target term are contained as a substring in the name or synonyms of the input term's name, synonym or definition. To provide a ranking score for retrieving candidate related terms, information content weighting was applied.

7.4.2.3 Relatedness between two terms

To evaluate the performance of the proposed method in calculating the relatedness between terms, a pseudo-negative dataset was built from random term pairs with the same number of term pairs as the positive dataset derived from the crossproducts. With a positive and a negative set of related pairs FLOR's ability to distinguish between related and unrelated pairs of terms can be tested. Three different setups of FLOR were applied to this dataset: (1) all descriptors, which considers all descriptors (names, definitions and synonyms) of GO terms; (2) just

names, where GO terms are described only by their names; and (3) without exact matches, that also describes GO terms using only their names and computes all scores using only Sim_{PM}.

The first two setups are intended to test the influence of considering the term definitions as well as names and synonyms. It has been suggested (Johnson *et al.*, 2006) that term definitions contain relevant information in the case of finding relations to ChEBI, so here we test if the same is true within GO. The third setup is designed to evaluate the influence of exact matches in the overall computation. With this setup FLOR's performance in distinguishing between positive and negative term pairs without recurring to exact matches can be tested.

To provide a basis of comparison, FiGO and a simple strict matching algorithm were also employed. The performance of all five methods was assessed through ROC and linear correlation analysis. ROC curves were plotted using the ROCR R library (Sing *et al.*, 2005).

Figure 7.4 presents the ROC curves for all five methods. All five methods have a good performance on this dataset, with a very low false positive rate which can ultimately be attributed to the randomly generated negative dataset. However, there is a clear advantage of the proposed method using all descriptors over FiGO, and FiGO is already a noteworthy improvement on strict matching. The linear correlation, presented in Table 7.3, corroborates the increased performance of two of FLOR's setups: all descriptors and just names.

Method	Pearson's correlation
FLOR *all descriptors*	0.89
FLOR *just names*	0.82
FLOR *without exact matches*	0.72
FiGO	0.73
strict matching	0.75

Table 7.3: Pearson's correlation values for the application of FLOR's 3 variants, FiGO and strict matching to GO term relatedness computation.

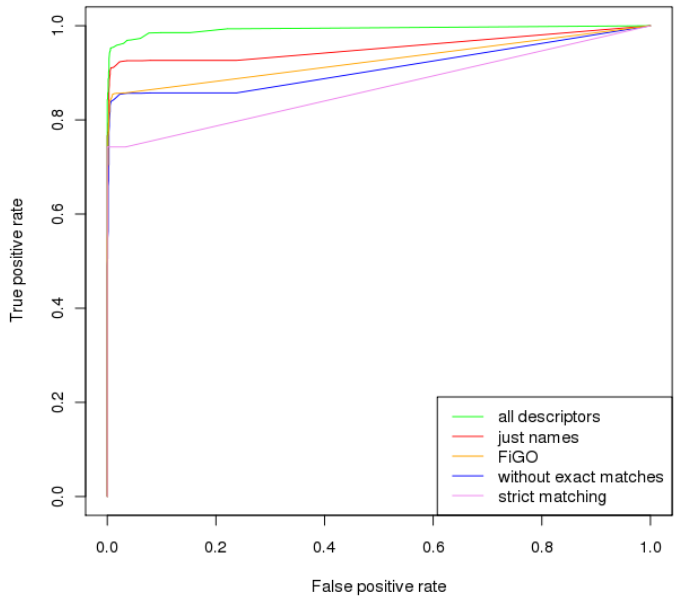

Figure 7.4: ROC curves for the comparison of five relation extraction methods

FLOR's all descriptors, just names and without exact matches setups, FiGO and strict matching.

7.4.2.4 Finding related concepts

GO crossproducts were also used to evaluate the performance of FLOR in retrieving terms related to an input term. For each term pair (query term - target term), a list of candidate terms that are related to the query term ranked according to their score is returned, and these are evaluate them against the target term. Precision and recall over all term pairs are calculated by considering the first k candidates. So, precision and recall at rank k are defined as follows:

$$precision(k) = \frac{\sum_{i=1}^{k} ||C^+||}{\sum_{i=1}^{k} ||C||} \tag{7.9}$$

$$recall(k) = \frac{\sum_{i=1}^{k} ||C^+||}{||N||} \tag{7.10}$$

where C^+ is the number of correct predictions, C is the number of candidates at that rank and N is the total number of term pairs.
However, these metrics are not adequate to evaluate the performance of the methods across all ranks. Gaudan *et al.* (2008) propose a global performance measure, *gp*, that is suitable for this evaluation. This measure is able to compare measures across all ranks, by weighting the contribution of correct predictions according to their rank. A correct prediction at rank 1, contributes fully, while a correct prediction at rank 10, contributes with a tenth. The formula is:

$$gp(k) = \frac{\sum_{i=1}^{k} 1/k||C^+||}{||N||} \tag{7.11}$$

Following this approach, FLOR's two best performing setups in the previous evaluation, all descriptors and just names, were evaluated against FiGO and strict matching, for ranks 1 through 10. Figure 7.5 shows these results. From the analysis of this data, it is clear that FLOR outperforms both FiGO and the strict matching approach. When considering all datasets, the best performance is achieved by the

all descriptors setup, with a precision of 0.42 at a recall of 0.42 in rank 1. When using just names, these values drop slightly to 0.40. The strict matching dataset is very close behind with 0.39 precision at 0.39 recall, whereas FiGO has a weaker performance, with 0.05 of precision/recall at rank 1.

When considering the global performance values given in Table 7.4, and inspecting each dataset's results separately, we can see clear different performance tendencies according to the GO hierarchy to which the target term belongs. Cellular component terms are overall the easiest to predict relations to, and they particularly stand out in the just names and strict matching results of the mf_x_cc and bp_x_cc datasets. This advantage is not as obvious in the all descriptors setup. Molecular function terms on the other hand are the most difficult to predict relations with, and show a clearly lower performance in the just names and strict matching, with some improvement in the all descriptors setup. The prediction of relations to biological process terms is very similar in the strict matching and just names, improving clearly with the all descriptors setup.

In terms of overall global performance, FLOR's all descriptors setup shows the best score, with 1.78, with just names coming in second at 1.69, and strict matching in third at 1.56. FiGO, while interestingly being the second best method in the bp_x_mf dataset, has the lowest global performance reaching only 0.33.

7.4.3 Discussion

The performance of FLOR in ontology enrichment was evaluated in two distinct tasks: calculating the relatedness between pairs of GO terms, and identifying candidate GO terms related to a query GO term. In the first task, all five methods tested have a good performance, being able to clearly distinguish between related and unrelated pairs of terms. However, the highest performers are FLOR's setups that consider both exact and partial matches. Since they both use exact and

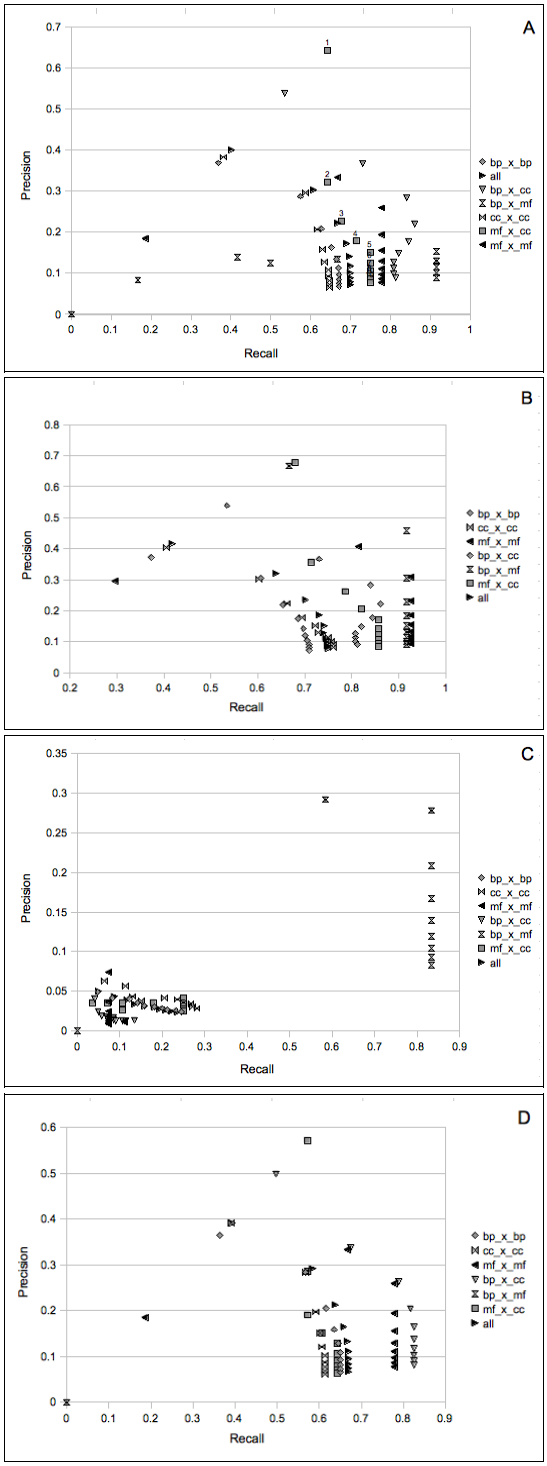

Figure 7.5: Precision vs. Recall for ranks 1 to 10. A -FLOR with the just names setup; B-FLOR with the all descriptors setup; C - FiGO; D - strict matching.

	strict matching	FiGO	just names	all descriptors
bp_x_bp	1.56	0.33	1.6	2.03
bp_x_cc	2.0	0.19	2.11	2.23
bp_x_mf	0.0	1.48	1.07	1.73
cc_x_cc	1.54	0.41	1.59	2.43
mf_x_cc	1.74	0.33	2	2.11
mf_x_mf	1.63	0.22	1.63	1.66
all	1.56	0.33	1.69	1.78

Table 7.4: Global performance of strict matching, FiGO and FLOR's just names and all descriptors setups.

partial matching approaches, these results support their integration.
In the second task, the FLOR setups were also the best performers presenting the highest global performance values, 1.78 and 1.69, followed by strict matching at 1.56 and with FiGO clearly lagging behind.
The strict matching method's performance is not surprising, since the GO crossproducts dataset is only composed of GO terms, which follow a standard of syntax and have a compositional nature, which increases the chances of a term being literally contained in another.
These same reasons also influence the good performance obtained by FLOR since it also employs an exact match approach. However, FLOR rises above the strict matching due to its hybrid nature capable of handling cases were an exact match is not possible. Furthermore, the superior performance of the all descriptors setup is a clear indicator that term definitions contain relevant information for relation extraction.
FiGO was found to be unsuited to this task. Its weak performance can be in part explained due to its ranking method that does not account for exact matches and filters out more general terms. Although these aspects are relevant to identify GO terms in abstracts, they can represent obstacles to the identification of relations between terms, where exact matches provide good evidence for the establishment of relations. Additionally, FiGO does not use term definitions, which can provide,

as already discussed, an important source of information.

However, the method applied is not the only factor that influences the results. There are obvious differences in terms of performance for the different ontologies of the target term. For instance, in the just names and strict matching results, cellular component terms appear to be more easily predicted to be in a relation than other branches terms. However, this advantage is not as strong in the all descriptors results, which can mean that while cellular component terms relations are stated in GO term's names, biological process and molecular function relations are more frequently referred to in the definition of terms, and thus are being missed by the just names and strict matching methods. In fact, 17% of all molecular function terms contain a cellular component term name in their names, and this percentage rises to 37% in the case of biological process terms. However, only 4% of biological process names contain molecular function names.

Nevertheless, cellular component terms may simply be easier to be identified in relations, since they are on average shorter than biological process and molecular function terms, and thus exact matches are more likely. A similar hypothesis has already been proposed by (Gaudan *et al.*, 2008), who observed that cellular component terms are the easiest to identify in text passages.

Overall these results highlight a relevant distinction that exists between finding an ontology concept in text, and retrieving related ontology concepts based on their textual descriptions: ontology textual resources are more rigidly defined than common natural language texts. Term names usually follow conventions, and term definitions are written in a clear and concise fashion, focusing on a very well defined piece of information. Natural language texts can refer to ontology terms in any number of manners, and usually possess a wider context that can increase the difficulty in identifying terms.

FLOR's ability to handle the natural variability in natural language while still regarding the more conventional format of ontology text data, make it particularly well suited to the task of retrieving related

ontology concepts, and thus capable of being applied to both ontology learning and ontology matching, and thus an appropriate method to support ontology extension.

7.5 Conclusions

In this chapter I described FLOR, a term relevance measure and its application in two scenarios: filtering term candidates extracted from text and finding relations between ontology concepts.

In candidate term filtering FLOR can be used after the application of automated term recognition methods to ensure that the candidate terms are within the domain of interest. The FLOR score can guide ontology developers in deciding whether a candidate concept is worth integrating into the ontology or not. This application scenario of FLOR can handle one of the issues in employing ontology learning methods to ontology extension, which is to ensure the relevance of the selected candidates. While testing the applicability of FLOR to candidate term filtering, I also investigated the feasibility of adding an extra filter based on named entity recognition. The intuition behind this, is that we can filter out irrelevant terms if we can identify them as entities belonging to other domains. An example of this was given using the GENIA tagger to weed out gene and cell mentions in the task of identifying novel GO terms in text.

FLOR was also shown to find relevant relations between Gene Ontology terms, supporting its application in this scenario. However, FLOR can also be applied within ontology matching to find the relatedness between concepts from distinct ontologies, as shall be described in the next chapter.

Future work in this area will include a broader evaluation of FLOR in ontology learning, investigating its ability to provide relevant candidate concepts.

However, FLOR has some limitations, namely that it depends on the

quality of the ontology vocabulary. For instance, if an ontology as few synonyms, two concepts can be related without sharing any words, rendering FLOR unable to detect them. The strategy using a named entity recognition system also has limitations, in particular, the existence of relevant NER systems for the domain being analyzed. If there are no NER systems able to identify the entities that need to be filtered out, then this approach cannot be applied.

FLOR's versatility has contributed to its application in other areas, namely the resolution of chemical entities using the ChEBI ontology Grego *et al.* (2012) and more recently to identifying epidemiology concepts in dictionary entries.

Chapter 8

Matching biomedical ontologies

Ontology matching is one of the core techniques in ontology engineering and can play an important role as a part of ontology development, since it is the basis for the integration of ontologies. By matching an ontology that needs extension to another relevant ontology, we can then reuse the matched portions for integration resulting in an extension of the ontology. Using ontology matching for extension is particularly interesting in the case of the biomedical domain, where despite the efforts of the community to provide orthogonal ontologies, it still boasts many overlapping ontologies or related resources. In BioPortal, a portal for biomedical ontologies, there are 306 ontologies distributed by categories, for instance 59 in the health category, 38 in the anatomy and 21 in the biological process.

In recent years the OAEI has been the major playfield for biomedical ontologies alignment, in its anatomy track. An important finding of the OAEI initiative is that many of the anatomy ontologies matches are rather trivial and can be found by simple string comparison techniques. Based on this notion, the work in (Ghazvinian *et al.*, 2009) has applied a simple string matching algorithm, LOOM, to several

ontologies available in the NCBO BioPortal, and reported high levels of precision in most cases. The authors mention several possible explanations for this, including the simple structure of most biomedical ontologies, the high number of synonyms they contain and the low language variability.

When compared to the top performers in OAEI 2008 (Caracciolo *et al.*, 2008), SAMBO, SAMBOdtf, and RiMOM, LOOM has a higher precision but a lower recall, probably due to the other algorithm's exploitation of other more complex methods than simple lexical matching. SAMBO and SAMBOdtf employ the UMLS (Unified Medical Language System) as the domain knowledge source to support lexical matching of concepts. RiMOM on the other hand, does not use external knowledge, relying on label and structural similarities.

In OAEI 2009 (Ferrara *et al.*, 2009), the best systems, SOBOM and AgreementMaker also did not use external knowledge, but both relied on global similarity computation techniques. These techniques represent ontologies as graphs, where concepts are nodes, and the relations between them, edges, and propagate lexical similarities between ontology concepts throughout the ontology graphs. This is based on the assumption that a match between two concepts should contribute to the match of their adjacent concepts, according to a propagation factor.

The importance of OAEI in the ontology matching field motivated my participation in 2010 and 2011 with methods developed within the context of this thesis. The OAEI provides a standard for the evaluation and comparison of ontology matching strategies and as such presents itself as an ideal benchmark for the testing of the proposed strategies. Moreover, it also supports an accurate analysis of results since the evaluation is based on a manually curated reference alignment.

Based on an analysis of winning systems of OAEI 2009 I developed methods designed to leverage on the success of simple lexical matching methods, while still finding alignments where lexical similarity is low, by using global computation techniques. These methods were

incorporated into a system called BLOOMS which was implemented as a module over the ontology matching platform AgreementMaker (Cruz & Sunna, 2008b) and was submitted to OAEI 2010. In 2011, the collaboration with the AgreementMaker team was tightened, and I was responsible for several improvements and new modules, which will be described below. These improvements were designed to handle and take advantage of the specific characteristics of biomedical ontologies, such as the richness in synonyms and the existence of a part_of hierarchy. In addition to testing these methods in OAEI I also investigated their performance on the alignment of a portion of GO and a portion of FMA, to support their usage on different biomedical ontologies scenarios.

8.1 Related Work

The LOOM system uses a terminological matcher that considers two concepts from different ontologies as similar, if their names or synonyms are equivalent based on a simple string-matching function, which disregards delimiters (e.g., spaces, underscores, parentheses, etc.), and allows for a one character mismatch in strings with length greater than four. The terminological matchers used in SAMBO comprise two approximate string matching algorithms, n-gram and edit distance, and a linguistic algorithm. The n-gram matcher, considers an n-gram as a set of n consecutive characters extracted from a string. If two strings share a high number of n-grams they are considered similar. Edit distance is defined as the number of deletions, insertions or substitutions required to transform one string into the other. If two string are easily transformed into each other, then they are considered similar.
Similarity flooding (simflood) (Melnik *et al.*, 2001) is a structural algorithm that is based on the notion that concepts from two graphs are similar if their adjacent elements are similar. The algorithm obtains initial matches using a string matching function, and then iteratively

creates more mappings for elements whose neighbors are similar. The contribution of similarity from a mapping to adjacent neighbors depends on the aligned node degree, i.e, if two aligned concepts have many parents, that each pair of parents gains little similarity, on the contrary if two aligned concepts have just one parent, they gain much more similarity. The iterations continue until the similarities between elements stabilize or a maximum number of iterations is reached.

The AgreementMaker system (Cruz & Sunna, 2008b) is an ontology matching framework that supports a wide variety of methods and matchers. Its architecture allows for serial or parallel composition of matchers such that the results from several matching algorithms can be combined into a single final result. Due to its modularity, AgreementMaker can be used in many different matching scenarios, but for the present purposes, only the strategies used in OAEI 2009 and 2010 will be described. For OAEI 2009 three string-based techniques were run on parallel: the Base Similarity Matcher (BSM) , the Parametric String-based Matcher (PSM), and the Vector-based Multi-word Matcher (VMM). BSM is a basic string matcher, which uses rule-based word stemming, stop word removal, and word normalization. PSM combines an edit distance measure and a substring measure. VMM uses the TF-IDF (Jones, 1972) approach by compiling a virtual document for every concept in an ontology and then uses the cosine similarity measure. The string matcher results are combined using the Linear Weighted Combination (LWC) matcher which automatically calculates the weights for each matcher using a local-confidence quality measure. Then AgreementMaker runs the Descendant's Similarity Inheritance (DSI) matcher, a structure-based matcher that increases the matching scores between the descendants of matched concepts. The final alignment is extracted from the set of matches based on a threshold for the scores. For OAEI 2010, the string-based matchers were extended by plugging in a set of lexicons that expands the set of

synonyms via WordNet. The final configuration used in the anatomy track was a Linear Weighted Combination of BSM, PMM and VMM.

8.2 Exploring lexical similarity and global computation techniques in OAEI 2010

The first ontology matching methods developed for this thesis were part of BLOOMS, a system that couples a lexical matching algorithm based on the specificity of words in the ontology vocabulary, with a novel global similarity computation approach that takes into account the semantic variability of edges.

8.2.1 Methods

BLOOMS has a sequential architecture composed of three distinct matchers: Exact, Partial and Semantic Broadcast Match. While the first two matchers are based on lexical similarity, the final one is based on the propagation of previously calculated similarities throughout the ontology graph. Figure 8.1 depicts the general structure of BLOOMS.

8.2.1.1 Lexical similarity

Exact and Partial matchers use lexical similarity based on textual descriptions of ontology concepts. Textual descriptors of concepts include their labels, synonyms and definitions. Since ontology concepts usually have several textual descriptors (e.g., name, synonyms, definitions), the similarity between two ontology concepts is given by the maximum similarity between all possible combinations of descriptors. The first matcher, Exact Match, is run on textual descriptions after normalization and corresponds to a simple exact match, where the score is either 1 or 0. The second matcher, Partial Match, is applied after processing all concept labels, synonyms and definitions through

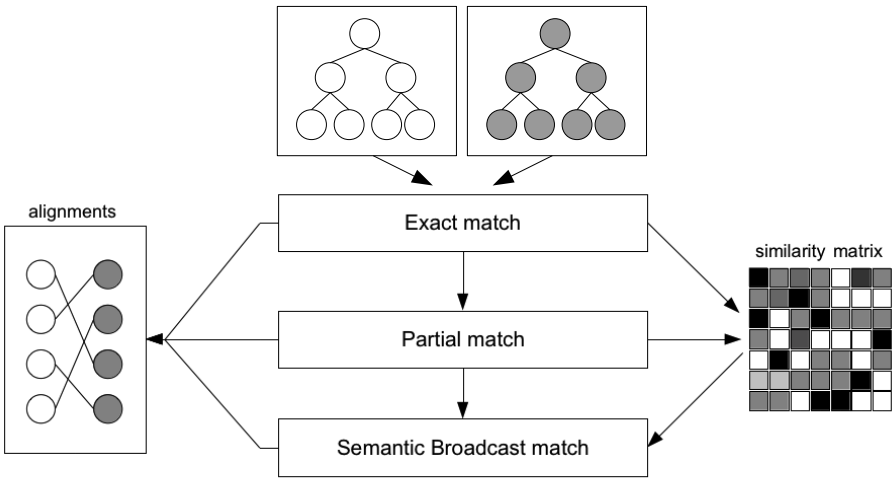

Figure 8.1: Diagram of BLOOMS architecture.

Given two ontologies, BLOOMS first extracts alignments based on Exact matches, then on Partial matches, and finally it propagates the similarities generated by those two strategies using the Semantic Broadcast approach.

tokenizing strings into words, removing stopwords, performing normalization of diacritics and special characters, and finally stemming (Snowball). If the concepts share some of the words in their descriptors, i.e. are partial matches, the final score is given by a Jaccard similarity, which is calculated by the number of words shared by the two concepts, over the number of words they both have.

These Exact and Partial matchers correspond to a specific configuration of FLOR, where in Sim_{EM} α is set to zero and in Sim_PM the EC of every word is set to one. Alternatively, each word can be weighted by its evidence content (see 7.2.2) and

to handle the cases where the word is common to both ontologies the evidence content of the word is given by the average of their ECs within each ontology.

8.2.1.2 Semantic Broadcast

After the lexical similarities are computed, they are used as input for a global similarity computation technique, Semantic Broadcast (SB). This novel approach takes into account that the edges in the ontology graph do not all convey the same semantic distance between concepts. This strategy is based on the notion that concepts whose relatives are similar should also be similar. A relative of a concept is an ancestor or a descendant whose distance to the concept is smaller than a factor d. To the initial similarity between concepts, SB adds the sum of all similarities of the alignments between all relatives weighted by their semantic gap sG, to a maximum contribution of a factor c. This is given by the following:

$$Sim_{final}(c_a, c_b) = Sim_{lex}(c_a, c_b) + c(\sum Sim_{lex}(r_i, r_j).sG(c_a, r_i, c_b, r_j)) \\ |D(c_a, r_i) < d \wedge D(c_b, r_j) < d \wedge r_i, r_j \in A \tag{8.1}$$

where c_a and c_b are concepts from ontologies a and b, and r_i and r_j are relatives of c_a and c_b at a distance D smaller than a factor d whose match belongs to the set of extracted alignments A. The semantic gap between two matches corresponds to the inverse of the average semantic similarity between the two concepts from each ontology. Several metrics can be used to calculate the similarity between ontology concepts, in particular, measures based on information content have been shown to be successful (Pesquita *et al.*, 2009b). BLOOMS implements three information content based similarity measures: Resnik (1998), Lin (1998) and a simple semantic difference between each concept's ICs. The information content of an ontology concept is a measure of its specificity in a given corpus. Many biomedical ontologies possess annotation corpora that are suited to this application. Nevertheless, semantic similarity can also be given by simpler methods based on edge distance and depth. Since neither the mouse or the human anatomy

ontologies have an annotation corpus, the Semantic Broadcast algorithm used a semantic similarity measure based on edge distance and depth, where similarity decreases with the number of edges between two concepts, and edges further away from the root correspond to higher levels of similarity.

Semantic broadcast can also be applied iteratively, with a new run using the similarity matrix provided by the previous.

8.2.1.3 Alignment Extraction

Alignment extraction in BLOOMS is sequential. After each matcher is run, alignments are extracted according to a predefined threshold of similarity and cardinality of matches, so that the concepts already aligned are not processed by matchers down the line. Each successive matcher has its own predefined threshold.

8.2.2 Integration in AgrementMaker to participate in OAEI 2010

BLOOMS was integrated into the AgreementMaker system due to its extensible and modular architecture. It was also of interest to benefit from its ontology loading and navigation capabilities, and its layered architecture that allows for serial composition since the BLOOMS approach combines two matching methods that need to be applied sequentially. Furthermore, the visual interface was also helpful during the optimization process of the matching strategy because it supports a very quick and intuitive evaluation.

Since neither the mouse or the human anatomy ontologies have an annotation corpus, the Semantic Broadcast algorithm used a semantic similarity measure based on edge distance and depth, where similarity decreases with the number of edges between two concepts, and edges further away from the root correspond to higher levels of similarity.

8.2.3 Results and Discussion

BLOOMS was evaluated in the OAEI anatomy track which consists of four tasks: in the first three tasks, matchers should be optimized to favor f-measure, precision and recall, in turn. In the fourth task, an initial set of alignments is given, that can be used to improve the matchers performance. In addition to the classical measures of precision, recall and f-measure, the OAEI initiative also employs recall+, which measures the recall of non-trivial matches, since in the anatomy track a large proportion of matches can be achieved using simple string matching techniques. Taking advantage of the SEALS platform several distinct configurations of BLOOMS were executed, testing different parameters and also analyzing the contribution of each matcher to the final alignment. It was found that after the first matcher is run, the alignments produced have a very high precision (0.98), but the recall is somewhat low (0.63). Each of the following matchers increases recall while slightly decreasing precision, which was expected given the increasing laxity they provide. It was also found that weighting the partial match score using word evidence content did not significantly alter results when compared to the simple Jaccard similarity. For task 1, a Partial Match threshold of 0.9 was used coupled with a final threshold of 0.4. Semantic Broadcast was run to propagate similarities through ancestors and descendants at a maximum distance of 2, and contribution was set to 0.4. This resulted in 0.954 precision, 0.731 recall, for a final F-measure of 0.828 and a recall+ of 0.315. For task 2, a Partial Match threshold of 0.9 was used and Semantic Broadcast was not run. With this strategy, we ensured a higher precision, of 0.967. However, recall was not much lower than the one in task 1, 0.725 , which resulted in a final f-measure of 0.829.

Comparing the results for tasks 1 and 2, they clearly indicate that the semantic broadcast strategy does not represent a very heavy contribution to recall. However when using both the Exact and Partial Match strategies, nearly 10% more matches are captured, than when

Table 8.1: OAEI 2010 anatomy track results

	Task #1			Task #2			Task #3			Recall+	
System	**Prec.**	**F**	**Rec.**	**Prec.**	**F**	**Rec.**	**Prec.**	**F**	**Rec.**	**#1**	**#3**
AgrMaker*	0.903	0.877	0.853	0.962	0.843	0.751	0.771	0.819	0.874	0.630	0.700
Ef2Match	0.955	0.859	0.781	0.968	0.842	0.745	0.954	0.859	0.781	0.440	0.440
NBJLM*	0.920	0.858	0.803	-	-	-	-	-	-	0.569	-
SOBOM	0.949	0.855	0.778	-	-	-	-	-	-	0.433	-
BLOOMS	0.954	0.828	0.731	0.967	0.829	0.725	-	-	-	0.315	-
TaxoMap	0.924	0.824	0.743	0.956	0.801	0.689	0.833	0.802	0.774	0.336	0.414
ASMOV	0.799	0.785	0.772	0.865	0.808	0.757	0.717	0.753	0.792	0.470	0.538
CODI	0.968	0.779	0.651	0.964	0.785	0.662	0.782	0.736	0.695	0.182	0.383
GeRMeSMB	0.884	0.456	0.307	0.883	0.456	0.307	0.080	0.147	0.891	0.249	0.838

using Exact Match alone. Recall+ is not very high, again highlighting the need to expand this strategy to improve recall. Nevertheless, performance was comparable to the best systems in 2009, and in 2010 the f-measure in task 1 is 5% lower than the best performing system, whereas in task 2 BLOOMS is the second best system, with a slight difference of 0.1% in precision. These are encouraging results which motivated the participation in OAEI 2011.

8.3 Exploring synonyms and other biomedical ontologies features in OAEI 2011

Following the participation in OAEI 2010, we decided to tighten our collaboration with the AgreementMaker team, and my contributions to the system focused on providing a detailed analysis of the anatomy tracks results alongside with designing several improvements to the matchers.

8.3.1 Analyzing the AgreementMaker alignments for OAEI 2010

A careful analysis of the 234 missed and 107 erroneous matches made by the AgreementMaker configuration used in OAEI 2010 revealed

several avenues for improvement. This analysis was only possible in 2011 due to the release of the reference alignment which was previously undisclosed.

Some of these mistakes were due to errors in the reference alignment or to peculiarities in the ontologies, for instance, in one ontology two terms correspond to two distinct classes, while in the other they are encoded as synonyms. Others were due to issues in the matchers themselves.

One very common problem was that matchers that take into account super and subclasses labels were only considering the is_a hierarchy to derive them, missing a lot of relevant information from the part_of hierarchy. In fact, there are 1669 part_of relations in the human anatomy ontology, whereas in the mouse there are 1637.

Another issue that was identified was that matchers were considering matches derived from synonym labels as good as matches derived from main labels, which also introduced some errors. Furthermore, some matches were being missed because the matchers were unable to identify their labels as synonyms. For instance, the match between the concepts MA:0002519 labeled 'stomach secretion' and NCI:C32661 labeled 'gastric secretion' was missed because the only synonym provided in the ontologies is 'gastric juice' in MA.

Some errors were due to matches between a concept and one of its subconcepts, for instance 'maxillary vein' to 'internal maxillary vein', or a concept and one of its parts, e.g. 'gut' to 'gut epithelium'.

Finally, a few errors were due to polysemic terms, such as 'lingula', which in MA is a part of the brain and in NCI a part of the lung.

These issues are due to ontologies features that are common to many other biomedical ontologies and as such present themselves as interesting avenues for developing improved matchers.

8.3.2 Methods

Following the analysis of AgreementMaker's results for 2010, a series of improvements were designed to handle some of the identified issues.

8.3.2.1 Extending the Ontologies with Synonym Terms

One of the improvements was a new method to augment the number of synonyms for each concept by capitalizing on the large number of synonyms present in both ontologies via the *hasRelatedSynonym* property. First, based on existing synonyms a lexicon of synonym terms (both single and multi-words) was derived. This is done by finding common terms between ontology synonyms to infer synonyms terms. Then these synonym terms are used to create novel synonyms, by substituting terms in existing synonyms and labels with their synonymous term. For example, in the mouse anatomy ontology the concept 'stomach serosa' has the synonym 'gastric serosa', which supports the inference that 'stomach' and 'gastric' are synonyms as well. These synonym terms can then be used to create new synonyms such as 'gastric secretion' for the concept 'stomach secretion', which allows it to be matched to its NCI counterpart.

8.3.2.2 Improving the Vector-based Multi-words Matcher

The VMM matcher compiles vectors of words from the concept's labels and then uses them to compute the similarity between concepts using a variety of metrics, such as TF-IDF coupled with cosine similarity, Jaccard similarity, Dice coefficient, etc. In AgreementMaker's classical implementation these vectors could be composed by the main labels, synonym labels and parent concepts labels. However, since the system wasn't considering the part_of hierarchy, it was missing these labels, so an option to use this hierarchy was added to the system. Another improvement was adding the option to also consider subclasses

labels.This broadens the number of labels available for the creation of the vectors which can be expected to have both an impact in precision and recall.
However, VMM was using a bag of words approach without regards to whether the labels were from a superclass, a subclass or a synonym. This could result in an erroneous calculation of the matching scores, for instance by comparing subclass words to superclass words.To handle this I implemented an alternative VMM matcher, called VMM Pairwise, which instead of using a single vector for all words in all labels, used several vectors, one for each label and compared them in a pairwise approach. The resulting score was given as the maximum similarity out of all pairwise similarities.

8.3.2.3 Other improvements

Two other contributions were also developed: a weighting for the BSM matcher and a best-match boosting strategy.
The weighting strategy enables the differentiation between matches based on main labels or synonym labels, by attributing a lower score to matches based on synonyms. This is achieved by multiplying the score by a factor. This strategy can handle the cases where some concepts are modeled as synonyms in one ontology and as sibling concepts in the other.

The best-match boosting strategy was developed to handle the cases of matches that were being missed due to scores below the threshold but that were in fact correct matches. Since these matches corresponded to the best match for both concepts, the strategy is to simply multiply the score by a factor.

8.3.2.4 Integration into AgreementMaker for OAEI 2011

The creation of the extended lexicon by the synonym terms approach impacts all lexical based matchers that AgreementMaker uses, BSM,

VMM and PSM. Likewise, the best-match boosting is also applied to all matchers. However, the use of the part_of hierarchy only affects VMM, and the weighting is exclusive to BSM. VMM Pairwise was not included in the competition configuration since although it improves on standard VMM in terms of performance, it was not contributing to an overall increase in performance when the other matchers were added (see Section 8.3.3).
For OAEI 2011 AgreementMaker used five matchers: BSM weighted, VMM, VMM considering the part_of hierarchy, PSM with the best-match boosting strategy and a Mediated Matcher (MM) based on alignments made to a mediating ontology. My contributions to MM were confined to which ontology was the best to use as a mediator. All matchers except MM used the synonym terms lexicon. The matchers results were then combined via LWC.

8.3.3 Results and Discussion

The results of AgreementMaker in the anatomy track are shown in Table ??. In 2011, OAEI focused on a single task to maximize f-measure and AgreementMaker ranked first with 91.7%, topping its own results for 2010 both in precision and recall. It is important to emphasize that improvements in performance will generally correspond to just a few percent, since the majority of matches are captured by simple string matching, a fact recognized by OAEI organizers.

To better highlight the performance of my contribution I provide a detailed analysis comparing the performance of the matchers before and after the inclusion of my improvements. Figure 8.2 provides a comparison for BSM with and without the synonyms terms and weighting approaches.
The BSM-allS matcher is equivalent to LOOM and presented as a baseline. The standard AgreementMaker BSM, here labeled as BSMlex-allS-LT, uses a lexicon and all available synonyms, providing an in-

Table 8.2: Results of the anatomy track in OAEI 2011

Matcher	Runtime	Size	Precision	F-measure	Recall	Recall+
AgrMaker	634	1436	.943	.917	.892	.728
LogMap	24	1355	.948	.894	.846	.599
AgrMaker_{2010}	-	1436	.914	.890	.866	.658
CODI	1890	1298	.965	.889	.825	.564
NBJLM_{2010}	-	1327	.931	.870	.815	.592
Ef2Match_{2010}	-	1243	.965	.870	.792	.455
Lily	563	1368	.814	.772	.734	.511
StringEquiv	-	*934*	*.997*	*.766*	*.622*	*.000*
Aroma	39	1279	.742	.679	.625	.323
CSA	4685	2472	.465	.576	.757	.595
MaasMtch	66389	438	.995	.445	.287	.003
MapPSO	9041	2730	.000	.000	.001	.000
MapEVO	270	1079	.000	.000	.000	.000
Cider	T	0	-	-	-	-
LDOA	T	0	-	-	-	-
MapSSS	X	0	-	-	-	-
Optima	X	0	-	-	-	-
YAM++	X	0	-	-	-	-

crease in recall of 5.5% to the expense of 2.7% precision. By adding the synonym terms extension and the weighing approach (BSMWlex-allS-ST) precision rises to 97.5% and recall rises to 71%. The weighting approach increases precision by 1.5%, which is a result of eliminating some erroneous matches that were being made to synonyms. The synonym terms approach increase recall by nearly 3% since it expands the number of synonyms available to support a match.

Table 8.3 presents the results for improvements to VMM and PSM. The best-match boosting strategy in PSM increases recall by 0.5% , and when it is incorporated into the complete matching strategy it raises f-measure by 0.4%. Boosting also improves recall in the VMM matcher by 2.3%. These improvements in recall are to be expected, since by raising the scores of reciprocal best matches these are raised above the threshold and make it to the final alignment. This means

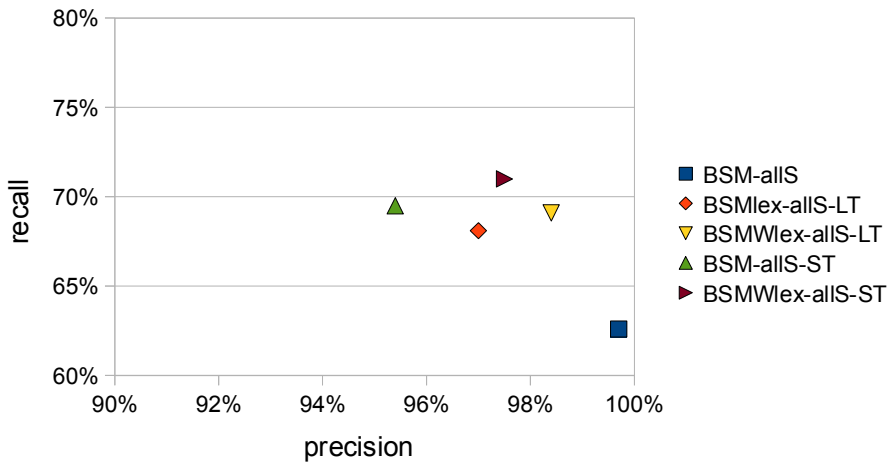

Figure 8.2: Alignment results for lexical matchers in the OAEI 2011 dataset.

allS: all synonyms; eS: exact synonyms; LT: lexicon terms; ST: synonym terms; W: weighted

that reciprocal best matches have a high probability of corresponding to matches even if they have scores slightly below the threshold.

Regarding the addition of the part_of hierarchy option to VMM, although results as a single matcher do not improve on the regular approach, it does have a positive impact on the complete matching strategy increasing recall by 2.4%. This is a result of finding a few new matches that VMM based on the is_a hierarchy was missing.

Although VMM Pairwise was not included in the configuration that was submitted to AgreementMaker, since it did not improve the overall performance, it represents an improvement over the standard VMM approach as seen in Table 8.4. The VMM Pairwise approach with the Jaccard similarity increases precision by nearly 20%, pushing f-measure up by more than 15%.

Table 8.3: Comparison of several matchers improved for OAEI 2011

Matcher	Precision	Recall	F-measure
Single matchers			
PSM-ST	59.1%	73.7%	65.6%
PSM-ST-boosted	59.1%	74.2%	65.8%
VMM	71.50%	67.00%	69.20%
VMM-ST	75.7 %	73.2%	74.4%
VMM-ST-boosted	75.1%	75.5%	75.3%
VMM-ST-part_of	73.50%	74.70%	74.10%
Complete matchers			
without improvements	95.4%	86.9%	90.9%
with ST	92.9%	89.1%	90.9%
with PSM boosted	96.6%	86.6%	91.3%
with VMM part_of	93.1%	89.3%	91.1%
with PSM boosted and VMM part_of	94.3%	89.2%	91.7%

Table 8.4: Comparative performance of VMM and VMM Pairwise

Matcher	Precision	Recall	F-measure
VMM-TFIDF	75.7%	73.2%	74.4%
VMM-ST-TFDIF	73.50%	74.70%	74.10%
VMM-ST-noSupClass-Jaccard	82.50%	68.00%	74.60%
VMMP-ST-noSupClass-TFIDF	80.7%	76.5%	78.5%
VMMP-ST-noSupClass-Jaccard	95.90%	71.00%	81.60%

8.4 Matching cellular component ontologies: GO and FMA

The development and improvement of matchers within the context of OAEI was specifically designed to handle the challenges of matching biomedical ontologies and as such their performance in other ontologies is expected to be positive. In the following, an evaluation of these matchers in the domain of cellular structures is given. However, due to the lack of gold standards for the alignment of other biomedical ontologies, a reference alignment had to be manually built. As such, this evaluation focuses on two subgraphs of two large ontologies: GO and FMA, rather than on the whole ontologies.

8.4.1 Reference Alignment

The reference alignment was built for two subgraphs of GO and FMA. The GO subgraph includes all descendants of the concept 'organelle' via is_a and part_of relations with a total of 149 concepts. The FMA subgraph includes all descendants of 'cellular component' via is_a and partonomy relations including part_of, general_part_of, constitutional_part_of and systemic_part_of, with a total of 816 concepts. The decision to align a smaller GO subgraph to a larger FMA subgraph was motivated by the fact that these ontologies have a distinct organization of concepts, and many of the concepts that are modeled as descendants of 'organelle' in GO, are modeled differently in FMA.
These subgraphs were manually aligned, resulting in a total of 41 matches.

8.4.2 Results and Discussion

The results for this alignment are shown in Table 8.5. One interesting difference between this alignment and the one for OAEI is that GO

has several different types of synonym properties such as *hasExactSynonym*, *hasRelatedSynonym*, *hasBroadSynonym* and *hasNarrowSynonym*. BSM without the lexicon plugin was used as a baseline to compare the influence between using all synonyms (BSM-allS) or just exact ones (BSM-eS). Using all synonyms, an extra match is found, but since it is incorrect it lowers precision. The addition of the lexicon plugin (BSM-eS-LT and BSM-aS-LT) provides no improvement. However, synonym terms (BSM-eS-ST and BSM-aS-ST) improves recall by 10 and 12%, with precision dropping a few percent. The best BSM matcher takes into account only exact synonyms, but uses both synonym terms and weighting (BSMW-eS-ST) for top values of both precision and recall with f-measure hitting 90.7%. This means that it was the improvements made to BSM in the context of this thesis that were responsible for an improvement of nearly 10% in f-measure.
Regarding VMM, using synonym terms resulted in an improvement of nearly 20% in f-measure, but performance was still low. However, when employing the pairwise approach proposed in this thesis there was nearly a 40% increase in f-measure to 85.3%. This supports the notion that a bag of words approach may not be well suited to handle ontologies with numerous synonyms such as GO. But in VMM, as in BSM, indiscriminately using synonyms regardless of their type lowers precision.

Since the reference alignment is small, some of the conclusions cannot be extrapolated. For instance, the use of all kinds of synonyms lowers precision without improving recall in this example, however it is expected that in larger ontologies this will not be the case. In fact, the more synonyms available the better the performance, as it can be seen when synonym terms are introduced. However, the improvement seen when using VMM pairwise instead of the standard bag of words approach, can be expected to translate well to the matching of larger ontologies.

Table 8.5: Comparison of matchers in the GO-FMA alignment

Matcher	Precision	Recall	F-measure
BSM-eS	100.00%	68.30%	81.20%
BSM-allS	96.60%	68.30%	80.00%
BSM-eS-LT	100.00%	68.30%	81.20%
BSM-aS-LT	96.60%	68.30%	80.00%
BSM-eS-ST	97.10%	80.50%	88.00%
BSM-as-ST	94.10%	78.00%	85.30%
BSMW-eS-ST (*a*)	100.00%	82.90%	90.70%
VMM-eS-LT-TFIDF	51.90%	34.10%	41.20%
VMM-eS-ST-TFIDF	55.60%	36.60%	44.10%
VMMP-aS-ST-Jaccard	65.3%	78.0%	71.1%
VMMP-eS-ST-Jaccard (*b*)	94.3%	80.5%	86.8%
PSM	45.80%	65.90%	54.00%
LWC(a,b)	91.90%	82.90%	87.20%
OAEI2010	87.10%	65.90%	75.00%

8.5 Conclusions

In this chapter I have presented contributions in ontology matching, specifically designed to address the challenges of matching biomedical ontologies. The first methods developed, and subsequently tested in OAEI 2010, were inspired by general findings of the state of the art and were shown to be good performers but unable to improve on the state of the art. Although global similarity techniques had produced good results in OAEI 2009, that was not the case with Semantic Broadcast. In fact, although AgreementMaker had used a global similarity technique for its 2009 entry, it was abandoned in 2010 in favor of more complex lexical matchers, supporting the conclusion that in this particular case their contribution is not very relevant.

However, when the methods were designed following a careful analysis of results, which was made possible by the release of the reference alignment for the OAEI 2011 competition, they had a significant contribution to the improvement of performance. Moreover the types of

issues found in the OAEI alignment and the avenues for their improvement were found to be generalizable to other biomedical ontologies, as shown in the alignment of two portions of GO and FMA.

One of the most meaningful contributions of this part of my work was the development of the synonym terms strategy. Although developed to handle an issue in matching, it is in fact an extension to the ontology in the form of new synonyms that is able to work without external resources.

Two of the identified issues in the OAEI 2010 alignment were not addressed directly in the improvements designed for the matchers in the context of this thesis: polysemes and matches to subclasses or parts. These represent interesting opportunities for future work. For instance, polysemous concepts can be filtered out by allowing the lack of similarity between neighbors to decrease the similarity between a pair of matched concepts. All of the methods developed here were independent of outside resources, however there are some matches that are virtually impossible to find without additional knowledge, so the next logical step would be to explore external resources.

Ontology matching can play an important role as a part of an ontology extension strategy, particularly in a domain such as biomedicine where there are many ontologies with overlapping domains. Using the alignments produced by ontology matching we can derive candidate concepts to include in the ontology to extend from the neighboring concepts of matched ones. The match scores can guide ontology developers in deciding on the suitability of the candidate concepts for integration into the ontology.

Part III

Conclusions

Chapter 9

Conclusions

9.1 Summary

This thesis addresses the theme of ontology extension in the context of biomedical ontologies. Its purpose was the development of a framework for semi-automated ontology extension as well as methods and methodologies that support the automation of some ontology extension processes in order to ease the burden on biomedical ontology developers. The proposed framework addresses the specific challenges of extending biomedical ontologies by providing three main components: extension prediction, learning and matching. The prediction extension component tackles the issues related to the large amounts of available scientific literature and related ontologies, by identifying the areas of the ontology in need of extension. These are used to focus the efforts of the matching and learning components which generate lists of candidate concepts by exploring the abundant biomedical literature and ontologies.

Given the existence of several systems for ontology learning and matching, the main focus of this thesis is the development of methods for the extension prediction component. I developed and tested two approaches: a rule based one and a supervised learning one. The rule based approach applies guidelines for the development of ontologies,

and it was shown to be unsuited for the task. The supervised learning approach relies on the notion that it is possible to learn a model that distinguishes between areas that will undergo extension and areas that will not, based on ontology features. This approach attained 0.79 of f-measure in predicting refinement for a subset of relevant biological process GO terms.

I also developed methods and techniques in both ontology learning and matching to ensure the success of these components in handling biomedical ontologies.

In the context of ontology learning a term relevance measure was developed to rank candidate concepts according to their relevance to the ontology domain. This is particularly relevant in highly specific and complex domains such as biomedicine. The proposed measure, FLOR, used the evidence content of words in the ontology vocabulary to measure the relatedness between existing and candidate concepts. FLOR was also applied to measure the relatedness between ontology terms to support new relations between them and to identifying ontology concepts in text.

Two sets of methods were developed for ontology matching and submitted to OAEI. The first, based on lexical similarity and global computation techniques, achieved good performance but ranked below state of the art systems. The second, resulting from the collaboration with the AgreementMaker developers, achieved the first place in the competition with an f-measure of 91.7% in aligning human and mouse anatomy ontologies. The methods employed in this resulted from a series of improvements on lexical matching methods that explored the specific issues and characteristics of biomedical ontologies. One of these methods was concerned with addition of new synonyms to the ontology and as such was not only a method for matching but also for extension. To test the application of these methods in another domain I aligned a portion of GO to a portion of FMA obtaining an f-measure value of 90.7%.

9.2 Research Contributions

In this thesis I contributed to the advancement of the state of the art in automated biomedical ontology extension by presenting a framework and methods for ontology extension specifically designed to address the issues of extending biomedical ontologies.

Framework: The proposed framework contemplates the need to adapt current extension methodologies to the biomedical domain, by integrating components of existing methods with specific methods developed in this thesis to provide a complete structure for a semi-automated ontology extension system.

Analyzing Ontology Extension: A conceptual framework for analyzing ontology extension was also developed to ensure that the subsequently developed methods for predicting extension were properly designed (Pesquita & Couto, 2011).

Analysis of Ontology Usage Patterns: I performed an initial a analysis of annotation versus extension to highlight some of the issues in ontology extension (Pesquita *et al.*, 2009c).

Prediction of Ontology Extension: The main contribution of this thesis is the development of methods for automating the first step in ontology evolution, which is a completely novel approach to support the extension of ontologies. I tested two approaches: a rule-based one that was shown to be ineffective (Pesquita & Couto, 2011) and a supervised learning one which obtained f-measure results between 60-80% in predicting the extension of the Gene Ontology (Pesquita & Couto, 2012).

FLOR: I developed a new term relevance measure and applied it to various tasks:

- Filtering candidate concepts - FLOR was used to measure the 'domainness' of candidate concepts to ensure their relevance to the ontology
- Finding relations between ontology concepts - this approach was used within the same ontology to enrich the ontology with relations, and between different ontologies to support their matching (Pesquita *et al.*, 2010)
- Resolution of ontology concepts extracted from text - FLOR was applied to the resolution of chemical concepts extracted from text to ChEBI concepts, increasing the f-measure by 2-5% over dictionary-based approaches (Grego *et al.*, 2012)
- Finding new ontology concepts in text - FLOR can also be applied in conjunction with a POS tagger to directly find candidate concepts in text

Analyzing the issues in matching biomedical ontologies: In this thesis I also analyzed the issues in aligning anatomical ontologies and identified them as common to the biomedical domain in general

Matching biomedical ontologies without external resources: I also contributed to the improvement of current state of the art ontology matching methods, tailoring them to handle the issues in aligning biomedical ontologies by exploring specific characteristics of these ontologies, such as synonyms and the part_of hierarchy (Pesquita *et al.*, 2010),(Cruz *et al.*, 2011). These methods achieved first place in an international competition for the alignment of biomedical ontologies with an f-measure of 91.7%. I also evaluated these methods in the alignment of portions of GO and FMA obtaining an f-measure of 90.7%.

A reference alignment for a portion of GO and FMA: I also manually constructed a reference alignment for portions of GO and FMA, which given the lack of such resources can be a valuable contribution for the community.

Extending an ontology with new synonyms: I developed a method for creating new synonyms for ontology concepts which functions both in the context of ontology matching, but also as a method for ontology enrichment (Cruz *et al.*, 2011).

9.3 Parallel contributions

During my PhD studies I also developed research, which although not being directly related to this thesis, contributed to it.

- A book chapter on using biomedical ontologies in mining biomedical information (Pesquita *et al.*, 2008b)
- A study of the coherence between manual and electronic annotations in GO Pesquita *et al.* (2009a).
- A tool for the collaborative evaluation of semantic similarity measures in GO Pesquita *et al.* (2009d).
- An approach for identifying bioentity recognition errors of rule-based text-mining systems (Couto *et al.*, 2008)
- A method citation metrics which handles self-citations using Google Scholar (Couto *et al.*, 2009)
- An international collaboration for the analysis of proteomics data using semantic similarity in the are of breast cancer research, which resulted in a publication in Nature Biotechnology (Taylor *et al.*, 2009)
- A study on molecular functions from a protein perspective using the Gene Ontology (Faria *et al.*, 2009)
- An analysis of proteomics data related to cystic fibrosis (Gomes-Alves *et al.*, 2010)
- A suite of tools for application of biomedical ontologies to the analysis of biomedical data(Tavares *et al.*, 2011)
- A methodology for gene identification using the Gene Ontology (Bastos *et al.*, 2011)

- A study on mining annotation data to highlight ontology design issues (Faria *et al.*, 2012)

9.4 Overall Approach

The methods described in this thesis can be integrated into a methodology following the proposed framework. Figure 9.1 presents the data flow diagram of this methodology.
In this thesis I developed methods for all three components of the framework. Using the ontology to extend as a source ontology, the Extension Prediction method generates a list of candidate ontology areas to extend. Then these areas are used as input to the processes Corpus Construction and Finding Relevant Ontologies. These processes were not addressed in this thesis since they rely on simple methods. For the Corpus Construction we can query PubMed or the Web with a composition of multiple queries using the names and synonyms of concepts in the areas to extend. Finding Relevant Ontologies can be accomplished by querying ontologies repositories such as BioPortal, or ontology search engines such as Swoogle also using the names and synonyms of concepts in the areas to extend.
The Relevant Corpus created in this fashion is used as input to Ontology Learning systems. Here, we can plugin any of the several available ontology learning systems that are able to output a list of candidate concepts. Then this list of candidate concepts is processed by the Candidate Filter, which is responsible for ensuring the relevance of the candidate concepts, by using the term relevance measure FLOR and/or the Named Entity filter.
The Relevant Ontology is used as input for the Ontology Matching process. Here any of the state of the art ontology matching systems can be used, but in this thesis we improved an existing system, AgreementMaker to handle the specific issues of matching biomedical ontologies. Candidate concepts are then extracted

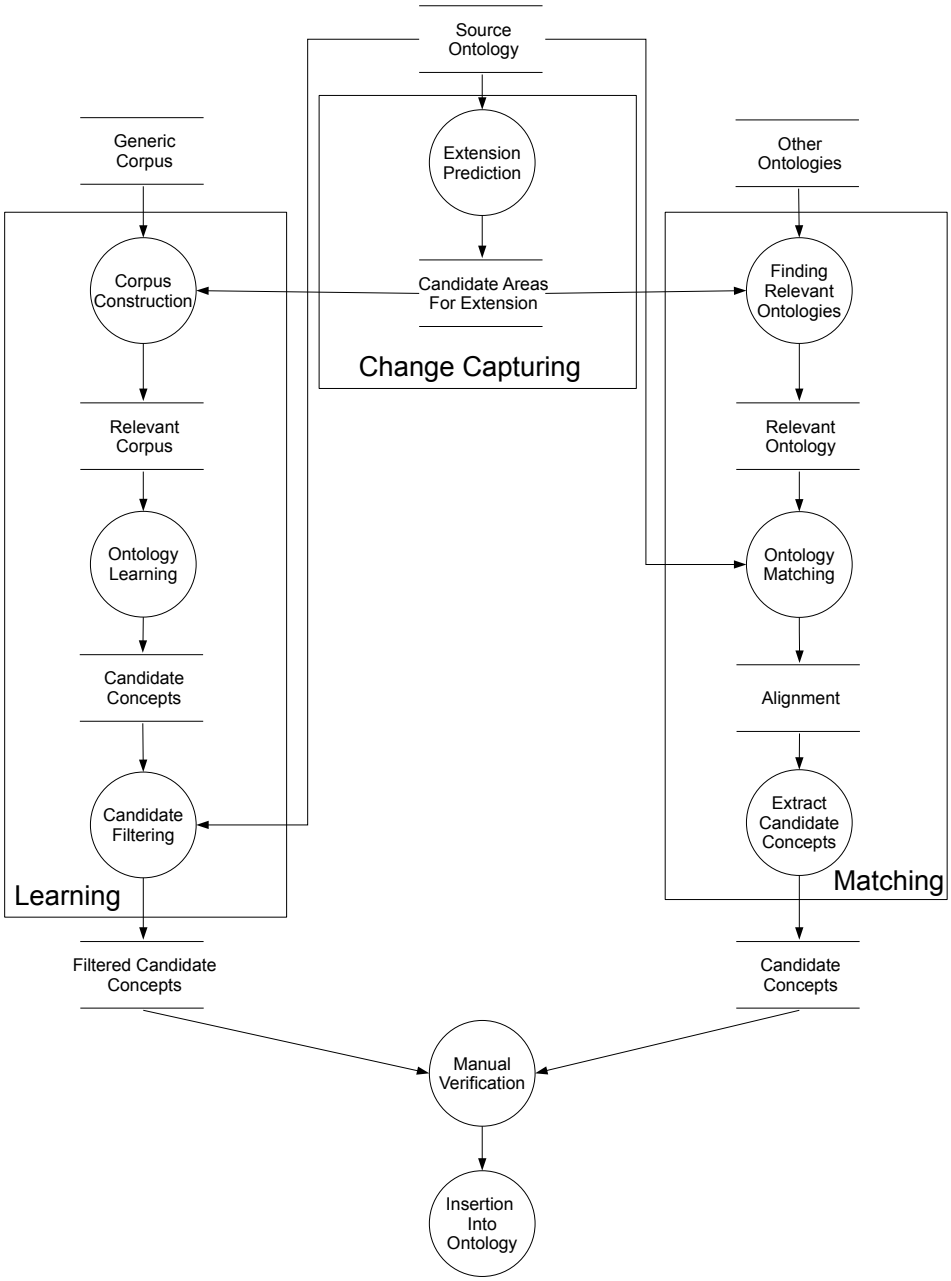

Figure 9.1: Data flow diagram of the proposed methodology.

from the produced alignment by retrieving the descendants of the matched concepts.
The candidate concepts lists are then manually verified by an expert and integrated into the ontology if considered valid. The scores returned by the Candidate Filter and by the Ontology Matching can help to guide ontology developers in this decision process.
Human experts will always be required either to verify the results of automated methods or to handle the more complex modeling tasks, but a clever integration of the proposed methods with traditional ontology development strategies is expected to dramatically decrease the workload of ontology engineers and drive the cost of ontology extension in the biomedical domain down, both in terms of time and resources.

The methods developed in this thesis can also be applied in other areas of ontology engineering. The prediction of ontology extension can be used to identify redundant areas of the ontology, i.e. areas that never change or are hardly ever used. This can be a contribution to ontology evolution, particularly in domains where ontologies model human created content as opposed to domains that model physical reality, and as such should only contain useful concepts. The methods in ontology matching can be employed in the matching of any ontologies even outside the biomedical domain, and should be particularly effective in ontologies that model synonyms but still fail to include all possible ones. Finally, the ranking and filtering methods in ontology learning can also be applied to other domains, and in particular the approach used in FLOR can be employed in ontologies with specific terminology whose concepts may be harder to identify in text.

9.5 Limitations and Future Work

Despite the success of the proposed methods and the validity of the overall approach, there are some limitations.

The extension prediction methodology depends on the existence of different versions of the ontology. Although versioning is a precept of OBO ontologies, many of the existing biomedical ontologies, particularly the smaller ones belonging to small development projects, have a single version. This does not limit however the application of the other components, which for instance can function with manually defined areas for extension. Another limitation of this methodology is that it can only predict the areas that will be extended by incremental knowledge acquisition since it is based on past events. It is not able to predict the need for extension generated by paradigm shifts (Kuhn, 1962), which although being a less frequent occurrence can lead to considerable changes in the underlying domain resulting in a need for redesigning the ontology. Two other limitations of the prediction methodology present themselves as interesting paths to be pursued in future work. On one hand, the proposed methodology does not contemplate the fact that ontology concepts are related, and as such machine learning algorithms should contemplate that the instances are not independent. On the other hand, when using several ontology versions to derive the features, they are temporally related, a fact that should also be addressed by the machine learning strategy.

The methods developed for filtering candidate concepts also have some limitations. FLOR relies on the ontology vocabulary to filter the candidates and as such it depends on the quality of the ontology vocabulary. If an ontology has few synonyms defined, a candidate concept and an ontology concept may be related and yet share no common words. In this case FLOR will fail to rank the candidate. The Named Entity filter depends on the existence

of NER systems for pertinent entities. These entities should be related to the domain in question, but not belong to it. The GENIA tagger is an interesting approach for use with the Gene Ontology, but in other domains, the choice of an appropriate NER system is crucial for the usefulness of this filter.

The ontology matching approaches developed in this thesis did not contemplate all identified issues, so these are obvious targets for future work. Polysemous concepts and matches to subclasses or parts can be addressed by methods that check for matches in ascendants or descendants to filter them out. Another issue in the developed methods is that they do not exploit external resources and these are crucial to correctly identify some matches, for which there is no support in the ontology information alone. This is a logical next step in future work in this area, and has in fact motivated a recently started FCT funded project, SOMER (Semantic Ontology Matching using External Resources[1]).

9.6 Final Remarks

In this thesis I addressed challenges in extending biomedical ontologies in a semi-automated fashion. Ontology extension in the biomedical domain is a very demanding task, given the sheer amount of new knowledge being produced on a daily basis. Automating some of the processes in ontology extension can represent a valuable contribution to both large and small ontologies. In the biomedical domain, both types of ontologies are common.

Large ontologies are usually maintained by larger teams and receive greater funding, but given their broad domain their extension still faces challenging issues. Keeping these ontologies up to date, with the smallest lag possible, entails considerable effort and resource investment. These ontologies benefit from all methods

[1] http://somer.fc.ul.pt

proposed in this thesis, particularly from the extension prediction methodology.
Smaller ontologies might not benefit from the extension prediction, since it is likely they need to be extended in general, but they can employ the other methodologies directly, without a need to focus efforts in the first place. Smaller ontologies usually have less resources and automating the extensions that can be found via learning and matching releases developers to focus on cornerstone modeling issues.

Biomedical ontologies play a progressively important role in bioinformatics, but also in the life sciences in general. As biology and health become increasingly data-driven, the need to add a semantic layer to these large collections of data becomes more pressing. Ontologies poise themselves as ideal to help explore, understand and extract relevant knowledge from the accumulated data. Applications of biomedical ontologies are becoming more prevalent, in such diverse areas as Rubin *et al.* (2008): search and query of heterogeneous biomedical data, data exchange among applications, information integration, Natural Language Processing, representation of encyclopedic knowledge and computer reasoning with data. But for an ontology to be truly successful and useful, it needs to be constantly developed and maintained. This a highly demanding task, especially in such an active and complex domain as the life sciences. In particular, keeping up with new knowledge is a challenge in this domain, so the automation of some of the processes in ontology extension can be extremely helpful.
Ontology extension systems still have limitations, particularly when it comes to extending large and complex domains. Most systems have only been tested on simple domains, using manually constructed corpora, and still have a suboptimal performance.
The contributions of this thesis are not only novel in their essence but also in the complexity of their target, representing therefore

valid approaches to handling the challenges of extending biomedical ontologies. Integrating the methods described here with already existing systems following the structure of the proposed framework results in a methodology for ontology extension that can be applied to biomedical ontologies or other ontologies with similar characteristics. Ontology extension is a crucial ontology engineering task and the automation of some of its processes can contribute not only to decrease the resource investment but also to ensure a timely update of the ontology, which can be crucial in rapidly evolving areas such as genomics, epidemiology or healthcare. I envision that the future ontology development will necessarily incorporate the automation of some of its processes, mainly those that are tedious and time-consuming, releasing ontology experts to focus on core modeling issues. It is the successful integration of human expertise and automated methods that will ensure that biomedical ontologies realize their full potential as essential tools for handling the challenges of knowledge management in the Life Sciences in the 21st century, to which I believe this thesis is a step forward.

References

Agirre, E., Ansa, O., Hovy, E. & Martinez, D. (2000). Enriching very large ontologies using the WWW. In *Proceedings of the ECAI Ontology Learning Workshop*. 36, 37, 43

Alfonseca, E. & Manandhar, S. (2002). Extending a lexical ontology by a combination of distributional semantics signatures. *Proc. of EKAW-2002*. 37, 39, 43

Alterovitz, G., Xiang, M., Hill, D.P., Lomax, J., Liu, J., Cherkassky, M., Dreyfuss, J., Mungall, C., Harris, M.A., Dolan, M.E., Blake, J.A. & Ramoni, M.F. (2010). Ontology engineering. *Nature biotechnology*, **28**, 2008–2011. 57

Ananiadou, S. & Mcnaught, J. (2006). Text Mining for Biology and Biomedicine. *Computational Linguistics*, 1–6. 59

Aranguren, M.E., Antezana, E., Kuiper, M. & Stevens, R. (2008). Ontology Design Patterns for bio-ontologies: a case study on the Cell Cycle Ontology. *BMC bioinformatics*, **9 Suppl 5**, S1. 2

Bastos, H., Tavares, B., Pesquita, C., Faria, D. & Couto, F. (2011). *Application of Gene Ontology to Gene Identification*. Springer. 157

Bendaoud, R., Toussaint, Y. & Napoli, A. (2008). PACTOLE: A methodology and a system for semi-automatically enriching an ontology from a collection of

texts. *Lecture Notes in Computer Science*, **5513**, 203–216. 40, 43

Bisson, G., Ndellec, C. & Caamero, D. (2000). Designing clustering methods for ontology building - the mok workbench. *In: Proc. ECAI Ontology Learning Workshop*, **7**. 30

Blaschke, C., Leon, E., Krallinger, M. & Valencia, A. (2005). Evaluation of BioCreAtIvE assessment of task 2. *BMC bioinformatics*, **6**, S16. 109

Bodenreider, O. & Stevens, R. (2006). Bio-ontologies: current trends and future directions. *World Wide Web Internet And Web Information Systems*, **7**, 256–274. 12

Brewster, C., Jupp, S., Luciano, J., Shotton, D., Stevens, R.D. & Zhang, Z. (2009a). Issues in learning an ontology from text. *BMC bioinformatics*, **10 Suppl 5**, S1. 53

Brewster, C., Jupp, S., Luciano, J., Shotton, D., Stevens, R.D. & Zhang, Z. (2009b). Issues in learning an ontology from text. *BMC bioinformatics*, **10 Suppl 5**, S1. 60

Buitelaar, P., Olejnik, D., Sintek, M., Technology, L. & Services, S.W. (2004). A Protégé Plug-In for Ontology Extraction from Text Based on Linguistic Analysis. *The Semantic Web: Research and Applications*, **5**. 30

Camon, E., Magrane, M., Barrell, D., Lee, V., Dimmer, E., Maslen, J., Binns, D., Harte, N., Lopez, R. & Apweiler, R. (2004). The Gene Ontology Annotation (GOA) Database : sharing knowledge in Uniprot with Gene Ontology. *Nucleic acids research*, **32**, 262–266. 14

Caracciolo, C., Hollink, L., Ichise, R., Meilicke, C., Pane, J. & Shvaiko, P. (2008). Results of the Ontology Alignment Evaluation Initiative 2008. *The 7th International Semantic Web Conference*. 36, 130

Ceusters, W. (2009). Applying evolutionary terminology auditing to the Gene Ontology. *Journal of biomedical informatics*, **42**, 518–29. 63

Ceusters, W. & Smith, B. (2006). A realism-based approach to the evolution of biomedical ontologies. *AMIA ... Annual Symposium proceedings / AMIA Symposium. AMIA Symposium*, 121–5. 63, 67

Chawla, N., Bowyer, K., Hall, L. & Kegelmeyer, W. (2002). SMOTE: synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, **16**, 321—-357. 76

Cimiano, P. & Staab, S. (2005). Learning concept hierarchies from text with a guided hierarchical clustering algorithm. In *Proc. Workshop on Learning and Extending Lexical Ontologies by using Machine Learning Methods at ICML 2005*. 60

Cimiano, P. & Völker, J. (2005). A framework for ontology learning and data-driven change discovery. *Proc. of the NLDB'2005*. 30

Cimiano, P., Maedche, A., Staab, S. & Volker, J. (2004). *Handbook on Ontologies*, chap. Ontology learning, 245–267. Springer Verlag, Berlin, Heidelberg. xix, 22, 24

Cimiano, P., Völker, J. & Studer, R. (2006). Ontologies on Demand? *Information Wissenschaft und Praxis*, **57**, 315–320. 26

Couto, F., Silva, M. & Coutinho, P. (2005). Finding genomic ontology terms in text using evidence content. *BMC bioinformatics*, **6**, S21. 109, 119

Couto, F., Grego, T., Pesquita, C., Bastos, H., Torres, R., Sanchez, P., Pascual, L. & Blaschke, C. (2008). Identifying bioentity recognition errors of rule-based text-mining systems. In *IEEE Third International Conference on Digital Information Management (ICDIM)*. 157

Couto, F., Pesquita, C. & Grego, T. (2009). SOMBRA: Semantic Ontology Matching based on Relationships and Argumentation. 157

Cruz, I., Stroe, C., Caimi, F., Fabiani, A., Pesquita, C., Couto, F. & Palmonari, M. (2011). Using agreementmaker to align ontologies for oaei 2011? *Ontology Matching*, 114. 5, 156, 157

Cruz, I.F. & Sunna, W. (2008a). Structural Alignment Methods with Applications to Geospatial Ontologies. *Transactions in GIS*, **12**, 683–711. 33, 36

Cruz, I.F. & Sunna, W. (2008b). Structural Alignment Methods with Applications to Geospatial Ontologies. *Transactions in GIS, Special Issue on Semantic Similarity Measurement and Geospatial Applications*, **12**, 683–711. 131, 132

Day-richter, J., Harris, M.A., Haendel, M., Obo-edit, T.G.O., Group, W. & Lewis, S. (2007). Databases and ontologies OBO-Edit — an ontology editor for biologists. *Bioinformatics*, **23**, 2198–2200. 15

Deane, P. (2005). A nonparametric method for extraction of candidate phrasal terms. *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics - ACL '05*, 605–613. 108

Denicola, a., Missikoff, M. & Navigli, R. (2009). A software engineering approach to ontology building. *Information Systems*, **34**, 258–275. 2

Euzenat, J. & Shvaiko, P. (2007). *Ontology matching*. Springer-Verlag New York Inc. 29, 31

Euzenat, J., Loup, D., Touzani, M. & Valtchev, P. (2004). Ontology alignment with OLA. In *Proceedings of the 3rd International Workshop on Evaluation of Ontology based Tools (EON), Hiroshima, Japan*. 33

Faatz, A. & Steinmetz, R. (2002). Ontology enrichment with texts from the www. *Semantic Web Mining 2nd Workshop at ECML/PKDD*. 38, 43

Faria, D., Pesquita, C., Couto, F. & Falcao, A.O. (2009). Goclasses: molecular function as viewed by proteins. In *BioOntologies SIG at ISMB/ECCB - 17th Annual International Conference on Intelligent Systems for Molecular Biology (ISMB)*. 157

Faria, D., Schlicker, A., Pesquita, C., Bastos, H., Ferreira, A.E., Albrecht, M. & Falcão, A.O. (2012). Mining GO annotations for improving annotation consistency. *PLoS ONE (in press)*. 158

Faure, D. & Nédellec, C. (1998). A corpus-based conceptual clustering method for verb frames and ontology acquisition. In *LREC workshop on adapting lexical and corpus resources to sublanguages and applications*, 707–728. 30

Ferrara, A., Hollink, L., Isaac, A., Joslyn, C., Meilicke, C., Nikolov, A., Pane, J., Shvaiko, P., Spiliopoulos, V. & Wang, S. (2009). Results of the Ontology Alignment Evaluation Initiative 2009. *Fourth International Workshop on Ontology Matching, Washington, DC*, **1**. 17, 36, 130

Flouris, G., Manakanatas, D., Kondylakis, H., Plexousakis, D. & Antoniou, G. (2008). Ontology change: classification and survey. *The Knowledge Engineering Review*, **23**, 117–152. 17, 19

Fortuna, B., Grobelnik, M. & Mladenič, D. (2006). Background knowledge for ontology construction. In *Proceedings of the 15th international conference on World Wide Web*, 950, ACM. 60

Frantzi, K., Ananiadou, S. & Mima, H. (1996). Automatic Recognition of Multi-Word Terms: the C-value/NC-value Method. *International Journal on Digital Libraries*. 108

Gaudan, S., Jimeno Yepes, a., Lee, V. & Rebholz-Schuhmann, D. (2008). Combining Evidence, Specificity, and Proximity towards the Normalization of Gene Ontology Terms in Text. *EURASIP journal on bioinformatics & systems biology*, **2008**, 342746. 122, 126

Gerstein, M., Seringhaus, M. & Fields, S. (2007). Structured digital abstract makes text mining easy. *Nature*, **447**, 142. 1

Ghazvinian, A., Noy, N. & Musen, M. (2009). Creating mappings for ontologies in biomedicine: Simple methods work. In *AMIA Annual Symposium (AMIA 2009)*, 2. 129

GO Consortium (2010). The Gene Ontology in 2010: extensions and refinements. *Nucleic acids research*, **38**, D331–5. 2, 12

Gomes-Alves, P., Couto, F., Pesquita, C., Coelho, A. & Penque, D. (2010). Rescue of f508del-cftr by rxr motif inactivation triggers proteome modulation associated with the unfolded protein response. *Biochimica et Biophysica Acta (BBA)-Proteins & Proteomics*, **1804**, 856–865. 157

Grego, T., Pesquita, C., Bastos, H. & Couto, F. (2012). Chemical entity recognition and resolution to chebi. *ISRN Bioinformatics*, **2012**. 5, 128, 156

Gruber, T. (2008). *Encyclopedia of Database Systems*, chap. Ontology. Springer-Verlag. 10

Gruber, T.R. (1993). Toward principles for the design of ontologies used for knowledge sharing. *in Formal Ontology in Conceptual Analysis and Knowledge Representation, N. Guarino and R. Poli, Editors*. 9

Haase, P. & Völker, J. (2005). Ontology learning and reasoning-dealing with uncertainty and inconsistency. In *Proceedings of the Workshop on Uncertainty Reasoning for the Semantic Web (URSW)*, 45–55, Springer. 28

Haase, P., Hotho, A., Schmidt-Thieme, L. & Sure, Y. (2005). Collaborative and usage-driven evolution of personal ontologies. *The Semantic Web: Research and Applications*, 125–226. 62

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P. & Witten, I. (2009). The WEKA data mining software: an update. *ACM SIGKDD Explorations Newsletter*, **11**, 10–18. 76

Hartung, M., Kirsten, T., Gross, A. & Rahm, E. (2009). OnEX: Exploring changes in life science ontologies. *BMC bioinformatics*, **10**, 250. 99

Hartung, M., Gross, A. & Kirsten, T. (2010). Discovering evolving regions in life science ontologies. *Data Integration in the Life Sciences*. 63, 99

Hayamizu, T.F., Mangan, M., Corradi, J.P., Kadin, J.a. & Ringwald, M. (2005). The Adult Mouse Anatomical Dictionary: a tool for annotating and integrating data. *Genome biology*, **6**, R29. 16

Huang, J., Dang, J., Huhns, M.N. & Zheng, W.J. (2008). Use artificial neural network to align biological ontologies. *BMC genomics*, **9 Suppl 2**, S16. 36

Isaac, A., Meilicke, C., Shvaiko, P., Hage, W.R.V. & Yatskevich, M. (2007). Results of the Ontology Alignment Evaluation Initiative 2007. *Evaluation*. 34

Jean-Mary, Y.R., Shironoshita, E.P. & Kabuka, M.R. (2009). Ontology matching with semantic verification. *Web Semantics: Science, Services and Agents on the World Wide Web*, **7**, 235–251. 34

Jimeno-Yepes, A., Berlanga-Llavori, R. & Rebholz-Schuhmann, D. (2009). Ontology refinement for improved information retrieval. *Information Processing & Management*. 41, 43

Johnson, H.L., Cohen, K.B., Baumgartner, W.A., Lu, Z., Bada, M., Kester, T., Kim, H. & Hunter, L. (2006). Evaluation of lexical methods for detecting relationships between concepts from multiple ontologies. *Pacific Symposium on Biocomputing. Pacific Symposium on Biocomputing*, 28–39. 120

Jones, K. (1972). A statistical interpretation of term specificity and its application in retrieval. *Journal of documentation*, **28**, 11–21. 132

Kageura, K. & Umino, B. (1996). Methods of automatic term recognition: A review. *Terminology*, **3**, 259–290. 107

Kavalec, M. & Svatek, V. (2005). A study on automated relation labelling in ontology learning. *Ontology learning from text: methods, evaluation and applications*, **123**, 44–58. 27

Korkontzelos, I., Klapaftis, I. & Manandhar, S. (2008). Reviewing and evaluating automatic term recognition techniques. *Lecture Notes in Computer Science*, **5221**, 248–259. 108

Krauthammer, M. & Nenadic, G. (2004). Term identification in the biomedical literature. *Journal of biomedical informatics*, **37**, 512–26. 25

Kuhn, T. (1962). *The structure of scientific revolutions*. University of Chicago press. 161

Laera, L., Tamma, V., Euzenat, J., Bench-Capon, T. & Payne, T. (2006). Reaching agreement over ontology alignments. *LNCSC*, **4273**, 371. 40

Lambrix, P. & Tan, H. (2006). SAMBO - system for aligning and merging biomedical ontologies. *Web Semantics: Science, Services and Agents on the World Wide Web*, **4**, 196–206. 33, 34, 36

Lee, J.B., Kim, J.j. & Park, J.C. (2006). Automatic extension of Gene Ontology with flexible identification of candidate terms. *Bioinformatics (Oxford, England)*, **22**, 665–70. 40, 43

Leenheer, P.D. & Mens, T. (2008). Ontology evolution: State of the Art and Future Directions. In M. Hepp, P. De Leenheer, A. de Moor & Y. Sure, eds., *Ontology Management for the Semantic Web, Semantic Web Services, and Business Applications, from Semantic Web and Beyond: Computing for Human Experience*, vol. 6, Springer. 61

Li, J., Tang, J., Li, Y. & Luo, Q. (2009). RiMOM: A Dynamic Multistrategy Ontology Alignment Framework. *IEEE Transactions on Knowledge and Data Engineering*, **21**, 1218–1232. 36

Lin, D. (1998). An information-theoretic definition of similarity. In *Proc. of the 15th International Conference on Machine Learning*, 296–304, Morgan Kaufmann, San Francisco, CA. 135

Liu, W., Weichselbraun, A., Scharl, A. & Chang, E. (2005). Semi-automatic ontology extension using spreading activation. *Journal of Universal Knowledge Management*, **1**, 50–58. 39, 43

Maedche, A. & Staab, S. (2000). Discovering conceptual relations from text. In *ECAI*, 321–325. 27

Maedche, A. & Staab, S. (2002). Measuring similarity between ontologies. *Lecture notes in computer science*, 251–263. 27

Maedche, A. & Staab, S. (2004). Ontology learning. *Handbook on Ontologies*, 173–190. 27

Mahn, M. & Biemann, C. (2005). Tuning co-occurrences of higher orders for generating ontology extension candidates. *Learning and Extending Lexical Ontologies by using Machine Learning Methods*, 28. 40, 43

Marquet, G., Mosser, J. & Burgun, A. (2007). A method exploiting syntactic patterns and the UMLS semantics for aligning biomedical ontologies: the case of OBO disease ontologies. *International journal of medical informatics*, **76 Suppl 3**, S353–61. 36

Melnik, S., Garcia-Molina, H. & Rahm, E. (2001). Similarity flooding: a versatile graph matching algorithm and its application to schema matching. *Proceedings 18th International Conference on Data Engineering*, 117–128. 32, 131

Miller, G. (1995). WordNet: a lexical database for English. *Communications of the ACM*, **38**, 41. 17

Mitchell, T. (1997). Machine learning. 1997. *Burr Ridge, IL: McGraw Hill*. 45

Morato, J., Marzal, M., Lloréns, J. & Moreiro, J. (2004). Wordnet applications. *Proceedings of GWC-2004*, 270–278. 17

Mungall, C.J., Bada, M., Berardini, T.Z., Deegan, J., Ireland, A., Harris, M.a., Hill, D.P. & Lomax, J. (2010). Cross-product extensions of the Gene Ontology. *Journal of biomedical informatics*. 117

Navigli, R. (2005). Semi-Automatic Extension of Large-Scale Linguistic Knowledge Bases. *Artificial Intelligence*. 37

Navigli, R. & Velardi, P. (2004). Learning Domain Ontologies from Document Warehouses and Dedicated Web Sites. *Computational Linguistics*, **30**, 151–179. 30, 108

Nováček, V., Laera, L., Handschuh, S. & Davis, B. (2008). Infrastructure for dynamic knowledge integration–automated biomedical ontology extension using textual resources. *Journal of biomedical informatics*, **41**, 816–28. 40, 42, 43

Noy, N.F. & Mcguinness, D.L. (2000a). Ontology Development 101 : A Guide to Creating Your First Ontology. *Development*, 1–25. 66

Noy, N.F. & Mcguinness, D.L. (2000b). Ontology Development 101: A Guide to Creating Your First Ontology. *Development*, 1–25. 10

Ogren, P., KB, C., Acquaah-mensah, G., Eberlein, J. & Hunter, L. (2004). The compositional structure of Gene Ontology terms. *Pac Symp Biocomput*, 214–225. 119

Pekar, V. & Staab, S. (2002). Taxonomy learning: factoring the structure of a taxonomy into a semantic classification decision. In *Proceedings of the 19th international conference on Computational linguistics*, 1–7, Association for Computational Linguistics, Morristown, NJ, USA. 38, 43

Pesquita, C. & Couto, F. (2012). Semi-automated extension of biomedical ontologies. *PLoS Computational Biology (in press)*. 5, 155

Pesquita, C. & Couto, F.M. (2011). Where GO is going and what it means for ontology extension. In *International Conference on Biomedical Ontologies*. 4, 5, 89, 90, 155

Pesquita, C., Faria, D., Bastos, H., Falcão, A.O. & Couto, F. (2008a). Metrics for go based protein semantic similarity: a systematic evaluation. *BMC Bioinformatics*, **9**. 111

Pesquita, C., Faria, D., Grego, T., Couto, F. & Silva, M. (2008b). *Untangling BioOntologies for Mining Biomedical Information*, 2009–2009. 157

Pesquita, C., Faria, D. & Couto, F.M. (2009a). Measuring coherence between electronic and manual annotations in biological databases. *Proceedings of the 2009 ACM symposium on Applied Computing - SAC '09*, 806. 15, 157

Pesquita, C., Faria, D., Falcão, A.O., Lord, P., Couto, F.M. & Bourne, P.E. (2009b). Semantic Similarity in Biomedical Ontologies. *PLoS Computational Biology*, **5**, e1000443. 4, 46, 135

Pesquita, C., Grego, T. & Couto, F. (2009c). Identifying Gene Ontology Areas for Automated Enrichment. In *Proceedings of the 10th International Work-Conference on Artificial*

Neural Networks: Part II: Distributed Computing, Artificial Intelligence, Bioinformatics, Soft Computing, and Ambient Assisted Living, 941, Springer. 155

Pesquita, C., Pessoa, D., Faria, D. & Couto, F. (2009d). Cessm : Collaborative evaluation of semantic similarity measures. In *JB2009: Challenges in Bioinformatics*. 157

Pesquita, C., Stroe, C., Cruz, I. & Couto, F. (2010). Blooms on agreementmaker: results for oaei 2010. *Ontology Matching*, 134. 5, 156

Pinto, H.S., Martins, P., Monte, B. & Pais, A.R. (1999). Some Issues on Ontology Integration. In *IJCAI-99 workshop on Ontologies and Problem-Solving Methods (KRR5)*, Borst 1997. 19

Pinto, H.S., Staab, S., Sure, Y. & Tempich, C. (2004). OntoEdit empowering SWAP: A case study in supporting DIstributed, Loosely-controlled and evolvInG Engineering of oNTologies (DILIGENT). *In C. Bussler, J. Davies, D. Fensel, and R. Studer, editors, First European Semantic Web Symposium, ESWS*, **pages**, 16–30. 2

Pinto, H.S., Tempich, C. & Staab, S. (2009). *Handbook on Ontologies*, chap. Ontology Engineering and Evolution in a Distributed World Using DILIGENT. Springer Berlin Heidelberg, Berlin, Heidelberg. 2

Porter, M. (2001). Snowball: A language for stemming algorithms. *URL http://snowball. tartarus. org/texts/introduction. html*. 110

Rada, R., Mili, H., Bicknell, E. & Blettner, M. (1989). Development and application of a metric on semantic nets. In *IEEE Transaction on Systems, Man, and Cybernetics*, 19, 17–30. 47

Rebholz-Schuhmann, D., Arregui, M., Gaudan, S., Kirsch, H. & Jimeno, A. (2008). Text processing through Web services: calling Whatizit. *Bioinformatics (Oxford, England)*, **24**, 296–8. 41

Resnik, P. (1998). Semantic Similarity in a Taxonomy: An Information-Based Measure and its Application to Problems of Ambiguity in Natural Language. *Journal of Artificial Intelligence Research*. 111, 135

Robertson, S.E. & Jones, K.S. (1976). Relevance weighting of search terms. *Journal of the American Society for Information Science*, **27**, 129–146. 108

Rosse, C. & Jr, L.V.M. (2003). A reference ontology for biomedical informatics : the Foundational Model of Anatomy. *Journal of Biomedical Informatics*, **36**, 478–500. 16

Rubin, D.L., Shah, N.H. & Noy, N.F. (2008). Biomedical ontologies: a functional perspective. *Briefings in bioinformatics*, **9**, 75–90. 1, 163

Ruiz-Casado, M., Alfonseca, E. & Castells, P. (2005). Using context-window overlapping in synonym discovery and ontology extension. In *Proceedings of RANLP-2005, Borovets, Bulgaria*. 39, 43

Sanchez, D. & Moreno, A. (2008). Learning non-taxonomic relationships from web documents for domain ontology construction. *Data & Knowledge Engineering*, **64**, 600–623. 27

Schmid, H. (1995). TreeTagger—a language independent part-of-speech tagger. 110

Schutz, A. & Buitelaar, P. (2005). Relext: A tool for relation extraction from text in ontology extension. *Lecture Notes in Computer Science*, **3729**, 593. 30

Sclano, F. & Velardi, P. (2007). Termextractor: a web application to learn the shared terminology of emergent web communities. In *Proc. of the 3rd International Conference on Interoperability for Enterprise Software and Applications I-ESA*, 28–30, Springer. 108

Seco, N., Veale, T. & Hayes, J. (2004). An intrinsic information content metric for semantic similarity in wordnet. In *ECAI*, 1089–1090. 47

Seddiqui, M.H. & Aono, M. (2009). An efficient and scalable algorithm for segmented alignment of ontologies of arbitrary size. *Web Semantics: Science, Services and Agents on the World Wide Web*, **7**, 344–356. 33

Shamsfard, M. (2004). Learning ontologies from natural language texts. *International Journal of Human-Computer Studies*, **60**, 17–63. 30

Sharan, U. & Neville, J. (2008). Temporal-Relational Classifiers for Prediction in Evolving Domains. In *Eighth IEEE International Conference on Data Mining, 2008. ICDM'08*, 540–549. 100

Sing, T., Sander, O. & Beerenwinkel, N. (2005). ROCR : Visualizing classifier performance in R. *Bioinformatics*, 2003–2004. 120

Sioutos, N., Coronado, S.D., Haber, M.W., Hartel, F.W., Shaiu, W.L. & Wright, L.W. (2007). NCI Thesaurus : A semantic model integrating cancer-related clinical and molecular information. *Journal of Biomedical Informatics*, **40**, 30–43. 16

Smith, B. (2003). *Blackwell Guide to the Philosophy of Computing and Information Blackwell Guide to the Philosophy of*

Computing and Information, chap. Ontology: An Introduction. Oxford: Blackwell. 10

Smith, B., Ashburner, M., Rosse, C., Bard, J., Bug, W., Ceusters, W., Goldberg, L., Eilbeck, K., Ireland, A., Mungall, C. *et al.* (2007). The obo foundry: coordinated evolution of ontologies to support biomedical data integration. *Nature biotechnology*, **25**, 1251–1255. 12

Stojanovic, L. (2004). Methods and tools for ontology evolution. *Mémoire de thèse, Université of Karlsruhe*. 74, 85

Stojanovic, L. & Motik, B. (2002). Ontology Evolution Within Ontology Editors. *Proceedings of the OntoWeb-SIG3 Workshop*, **pp**, 53–62. 18, 66

Stojanovic, L., Maedche, A., Motik, B. & Stojanovic, N. (2002). User-driven Ontology Evolution Management. *Proceedings of the 13 th International Conference on Knowledge Engineering and Knowledge Management (EKAW-02), Lecture Notes in Computer Science (LNCS), Volume 2473, Springer-Verlag*, **pp**, 285–300. 18, 61, 63, 95

Sure, Y., Staab, S., Studer, R. & Gmbh, O. (2003). *Handbook on Ontologies, International Handbooks on Information Systems*, chap. On-to-knowledge methodology (OTKM), 117–132. Springer-Verlag. 2

Tavares, B., Bastos, H., Faria, D., Ferreira, J.D., Grego, T., Pesquita, C. & Couto, F. (2011). The biomedical ontology applications (boa) framework. In *Proceedings of ICBO*. 157

Taylor, I.W., Linding, R., Warde-Farley, D., Liu, Y., Pesquita, C., Faria, D., Bull, S., Pawson, T., Morris, Q. & Wrana, J.L. (2009). Dynamic modularity in pro-

tein interaction networks predicts breast cancer outcome. *Nature biotechnology*, **27**, 199–204. 157

Tsuruoka, Y., Tateishi, Y., Kim, J.d. & Ohta, T. (2005). Developing a Robust Part-of-Speech Tagger for Biomedical Text. *Advances in Informatics*, 382–392. 108

Velardi, P., Fabriani, P. & Missikoff, M. (2001). Using text processing techniques to automatically enrich a domain ontology'. *in Proceedings of the ACM International Conference on Formal Ontology in Information Systems*. 37, 43

Velardi, P., Navigli, R., Cucchiarelli, A. & Neri, F. (2003). Evaluation of OntoLearn, a methodology for automatic learning of domain ontologies. *Ontology Learning from Text: Methods, evaluation and applications*, 1–15. 27

Völker, J., Haase, P. & Hitzler, P. (2008). Learning expressive ontologies. In *Proceeding of the 2008 conference on Ontology Learning and Population: Bridging the Gap between Text and Knowledge*, 45–69, IOS Press. 28

Wächter, T., Schroeder, M. & Dresden, T.U. (2010). Semi-automated ontology generation within OBO-Edit. *Bioinformatics*, **26**, 88–96. 41, 43

Weichselbraun, A., Wohlgenannt, G., Scharl, A., Granitzer, M., Neidhart, T. & Juffinger, A. (2009). Discovery and evaluation of non-taxonomic relations in domain ontologies. *International Journal of Metadata, Semantics and Ontologies*, **4**, 212. 28

Wermter, J. & Hahn, U. (2005). Finding new terminology in very large corpora. *Proceedings of the 3rd international conference on Knowledge capture - K-CAP '05*, 137. 108

Widdows, D. (2003). Unsupervised methods for developing taxonomies by combining syntactic and statistical information. *In Proceedings of the Human Language Technology Conference/North American Chapter of the Association for Computational Linguistics*, 276–283. 38, 43

Wilbur, J., Smith, L. & Tanabe, L. (2007). Biocreative 2. gene mention task. In *Proceedings of the second biocreative challenge evaluation workshop*, vol. 23, 7–16. 114

Witschel, H. (2005). Using decision trees and text mining techniques for extending taxonomies. *Learning and Extending Lexical Ontologies by using Machine Learning Methods*, 61. 38, 43

Zhang, S. & Bodenreider, O. (2007). Lessons Learned from Cross-Validating Alignments between Large Anatomical Ontologies. *MedInfo*, **12**, 822–826. 34

Zhang, S., Bodenreider, O. & Services, H. (2005). Alignment of Multiple Ontologies of Anatomy: Deriving Indirect Mappings from Direct Mappings to a Reference. *Symposium A Quarterly Journal In Modern Foreign Literatures*, 864–868. 17

Zhang, Z., Iria, J., Brewster, C. & Ciravegna, F. (2008). A comparative evaluation of term recognition algorithms. In *Proceedings of the Sixth International Language Resources and Evaluation (LREC'08)*. 108

Zhou, L. (2007). Ontology learning: state of the art and open issues. *Information Technology and Management*, **8**, 241–252. 22, 29