Advancing Ontology Alignment:
New Methods for Biomedical Ontology Alignment
using Non Equivalence Relations

by

Suzette Kruger Stoutenburg

B.Sc., Rowan State University, 1985

M.Sc., National Technological University, 1989

A dissertation submitted to the Graduate Faculty of the
University of Colorado at Colorado Springs

in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

Department of Computer Science

2009
This dissertation for
Doctor of Philosophy Degree by

Suzette Kruger Stoutenburg

has been approved for the
Department of Computer Science
by

Dr. Jugal Kalita, Chair

Dr. Marijke F. Augusteijn

Dr. Terrance Boult

Dr. Lisa Hines

Dr. Leo Obrst

Dr. Sudhanshu K. Semwal

Dr. Rex Welshon

Dr. Xiaobo Zhou

Date
Advancing Ontology Alignment: New Methods for Biomedical Ontology Alignment using Non Equivalence Relations

Thesis directed by Professor Jugal K. Kalita

Increasingly, ontologies are being developed and exposed on the Web to support a variety of applications, including biological knowledge sharing, enhanced search and discovery, and decision support. This proliferation of new Web knowledge sources is resulting in a growing need for integration and enrichment of these sources. Automated and semi-automated solutions to aligning ontologies are emerging that address this growing need with very promising results. However, nearly all approaches have focused on aligning ontologies using relationships of similarity and equivalence and none have applied knowledge in upper ontologies. None to our knowledge have applied Support Vector Machine (SVM) technology. Only very recently, solutions for scalability of ontology alignment have begun to emerge.

The goal of this research is to advance the state of the art in automated ontology alignment by contributing in three main areas. First, we present new algorithms to extend the information that can be derived in ontology alignment; specifically, new relationships between ontological components beyond similarity and equivalence. In particular, we present algorithms to align ontologies using subclass, superclass and relations contained within the original ontologies. We show how ontology alignment can be modeled in a Support Vector Machine and that use of SVMs enhances the ontology alignment process. Second, we contribute new evidence for ontology alignment.
We show that the use of semantics in conjunction with upper ontologies and other linguistic resources can enhance the alignment process and specifically contribute to the discovery of new relationships cross-ontology. Finally, we investigate scalability issues in the area of processing, reasoning, and aligning large-scale ontologies. We present an alignment algorithm that bounds processing by selecting optimal subtrees to align and show that this significantly improves efficiency without major reduction in precision.
To my husband and best friend, Greg

To my parents, Mildred and the late Karl Kruger
Acknowledgments

What a relief and thrill to be finally writing this section! This has been a long journey and it is great to finally arrive.

First, I would like extend a special thanks to my advisor, Dr. Jugal Kalita, for his outstanding support and guidance throughout this process. His technical depth, enthusiasm and drive kept me going. Dr. Kalita gave me great freedom to pursue areas of interest to me and yet also provided a wealth of ideas on how to improve the approach. I might add that it was also a pleasure to watch his daughter Ananya grow before my eyes, from a baby into an adorable child, as we met week after week to review the research.

I would like to acknowledge and thank my committee members for their ideas, support and guidance: Dr. Marijke F. Augusteijn, Dr. Terrance Boult, Dr. Lisa Hines, Dr. Sudhanshu K. Semwal, Dr. Rex Welshon and Dr. Xiaobo Zhou. Multiple exchanges with these outstanding educators enhanced the quality of the research and enriched my experience along the way. Ms. Patricia Rea also deserves a special thanks, as she was always very responsive and supportive throughout the process.

A special thanks is extended to Dr. Lisa Hines and Kaily Ewing for their support and collaboration in developing reference alignments for evaluation of alignment in the biomedical domain. Dr. Hines and Ms. Ewing also provided expertise and ideas for how to apply this research in the future in the biomedical domain. Their expertise was invaluable to this work. I would like to thank Justin Gray for choosing to do his Master’s thesis in ontology alignment. His software engineering expertise
and contributions contributed to this research early on in the process. I would also like to thank the members of Dr. Kalita’s study group, including Jeremy Witmer, Justin Gray, Michael Bankston, Beaux Shariff, Mona Habib and others. Their ideas and support at our weekly meetings were very helpful to this research. I also thank Melissa Garner for her invaluable help in advanced statistics.

I want to acknowledge the MITRE Corporation for offering me this tremendous opportunity. For many years, MITRE has remained one of Fortune’s 100 Best Companies to Work For, in part, because of the corporation’s commitment to education. I remain honored to be part of such an outstanding organization. Special thanks go to my colleagues at MITRE for their support and encouragement during this four year process. There are too many to mention and I don’t want to leave anyone out. There were many times when my colleagues had to fill in for me, taking on significant extra work during my absences. I greatly appreciate all that they did to help me succeed in this endeavor and I am forever in their debt. In addition, I received continuous support from ontology and alignment experts across MITRE. Their expertise and willingness to help is also greatly appreciated.

I would like to extend a special thanks to my colleague, friend and research mentor, Dr. Leo Obrst. Leo was the original inspiration for my taking on this mid-life challenge. Leo and I collaborated together on a research proposal in 2004; soon after, I started this Ph.D. program with the initial goal of being able to follow Leo’s technical discussions in the areas of logic and semantics. I believe I may have achieved that goal and so much more, thanks to Leo’s encouragement and unending willingness to share his expertise and spend time educating others.
I would also like to thank my mother and dear friend, Mildred and my late father Karl. Without their unconditional love and support, I would not have been equipped to take on this challenge. They taught me to be a strong, independent woman and that education was the path to success and freedom. My mom also deserves special thanks for always being there to help us and for putting up with my long absences, particularly in the last few months of the dissertation preparation.

Finally, I would like to thank my husband Greg for his unwavering support and encouragement during this time. Greg served as my IT support, psychologist and confidant. He fixed internet outages, debugged network problems and configured Virtual Machines with a moment’s notice and never a complaint. He gave me endless encouragement during tough times, late nights and 90+ hour weeks. He made sure we were stocked with light bulbs and caffeine sources, not to mention batteries for the wireless mouse and keyboard, especially in the final weeks of writing. His dedication to me and our family kept operations in our home moving (not only IT, but grocery shopping, bill paying, dog washing, cooking, vacuuming, etc.) and his belief in me was my primary inspiration. His patience with me and ability to make me laugh when all seems lost is legendary. We truly share in this accomplishment.
Table of Contents

List of Tables xv

List of Figures xvii

1 Introduction 1
   1.1 Background 2
   1.2 Motivation 5
   1.3 Summary of Research Goals and Contributions 7
   1.4 Research Methodology 9
   1.5 Organization of Dissertation 11

2 Definitions 13
   2.1 Ontology 13
   2.2 Representation of Ontologies 17
   2.3 Ontology Alignment 20
   2.4 Relationships in Ontology Alignment 23
   2.5 The Process of Ontology Alignment 25

3 State of the Art in Ontology Alignment 28
   3.1 Introduction 28
   3.2 Alignment Techniques over Schemas 29
   3.3 Alignment Techniques over Instances 31
   3.4 Alignment Techniques over Ontologies 32
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.2.1 Representation of Evidence Primitives and Features in a Support Vector Machine</td>
<td>95</td>
</tr>
<tr>
<td>6.2.2 Features to Support Ontology Alignment</td>
<td>98</td>
</tr>
<tr>
<td>6.2.3 Architecture for Ontology Alignment</td>
<td>100</td>
</tr>
<tr>
<td>6.3 Evaluation</td>
<td>101</td>
</tr>
<tr>
<td>6.4 Experimental Results</td>
<td>105</td>
</tr>
<tr>
<td>6.5 Conclusions</td>
<td>109</td>
</tr>
<tr>
<td>7 Scaling Ontology Processing, Reasoning and Alignment</td>
<td>112</td>
</tr>
<tr>
<td>7.1 Scalability in Ontology Processing and Reasoning</td>
<td>113</td>
</tr>
<tr>
<td>7.2 Scalability of Ontology Alignment</td>
<td>117</td>
</tr>
<tr>
<td>7.2.1 Approaches to Scaling Alignment</td>
<td>118</td>
</tr>
<tr>
<td>7.2.2 Partitioning and Thresholding</td>
<td>120</td>
</tr>
<tr>
<td>7.3 Metrics for Large-Scale Ontology Alignment</td>
<td>123</td>
</tr>
<tr>
<td>7.4 A Scalable Infrastructure for Ontology Processing and Alignment</td>
<td>124</td>
</tr>
<tr>
<td>7.4.1 Infrastructure for Ontology Processing</td>
<td>125</td>
</tr>
<tr>
<td>7.4.2 Optimizing Feature Extraction in Ontology Alignment</td>
<td>127</td>
</tr>
<tr>
<td>7.4.3 Optimizing Ontology Alignment with Partitioning</td>
<td>129</td>
</tr>
<tr>
<td>7.4.4 Optimizing Ontology Alignment with a Branch and Bound Approach</td>
<td>134</td>
</tr>
<tr>
<td>7.5 Experimental Results: Alignment of Large Scale Biomedical Ontologies</td>
<td>140</td>
</tr>
<tr>
<td>7.6 Conclusions</td>
<td>145</td>
</tr>
<tr>
<td>8 Conclusions and Future Research Directions</td>
<td>148</td>
</tr>
</tbody>
</table>

xiii
8.1 Summary and Conclusions ............................................. 148

8.2 Continuing Challenges in Ontology Alignment .................. 151
  8.2.1 Difficulty of the Alignment Problem ......................... 151
  8.2.2 Evaluation Issues ............................................. 153
  8.2.3 Metrics Issues .................................................. 155
  8.2.4 Scalability Issues .............................................. 157

8.3 Future Research Directions ......................................... 158
  8.3.1 Alignment in the Biomedical Domain ......................... 158
    8.3.1.1 Enhancing Data Analysis for Biological Researchers . 159
    8.3.1.2 Hypothesis Generation for Human Disease Research 162
    8.3.1.3 Complementing Gene Expression Clustering Techniques163
  8.3.2 Alignment for Rapid Enterprise Integration ................. 164
  8.3.3 Alignment for Web Service Composition and Mediation .... 166
  8.3.4 Role-based Ontology Alignment .............................. 171
  8.3.5 Additional Directions ......................................... 172

A Reference Alignment Examples in the Academic Domain 174

B Reference Alignment Examples in the Biomedical Domain 179

C Evidence Primitives for Ontology Alignment 183
List of Tables

4.1 Academic Ontologies Used in Performance Evaluation .......................... 44
4.2 Biomedical Ontologies Used in Performance Evaluation ......................... 44
4.3 Relationship pattern frequency in Academic ontologies used in this work 48
4.4 Relationships in Open Biomedical Ontologies (OBO) ........................... 52
5.1 Evidence primitive types ........................................................................ 57
5.2 Evidence primitives, sources and set membership ................................. 58
5.3 Subclass Pattern 1 Evaluation Results .................................................. 68
5.4 Subclass Pattern 2 Evaluation Results .................................................. 71
5.5 Subclass Pattern 3 Evaluation Results .................................................. 74
5.6 Superclass Pattern Evaluation Results .................................................... 79
5.7 Evaluation of Inferred Relation Patterns ............................................... 82
6.1 Performance of alignment of academic ontologies, under a variety of test conditions ................................................................. 106
6.2 Performance of SVM in alignment of the GO and MP ontologies ............... 107
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.3</td>
<td>Performance of SVM in alignment of the GO and MP ontologies using</td>
<td>109</td>
</tr>
<tr>
<td></td>
<td>specialized heuristics.</td>
<td></td>
</tr>
<tr>
<td>7.1</td>
<td>Size comparison of large biomedical ontologies and ontologies used in</td>
<td>114</td>
</tr>
<tr>
<td></td>
<td>the Ontology Alignment Evaluation Initiative (OAEI) 2008.</td>
<td></td>
</tr>
<tr>
<td>7.2</td>
<td>Loading time for large-scale biomedical ontologies.</td>
<td>128</td>
</tr>
<tr>
<td>7.3</td>
<td>Alignment Evaluation Results for Large Scale Biomedical Ontologies.</td>
<td>144</td>
</tr>
<tr>
<td>8.1</td>
<td>Proposed set of ontology alignment services</td>
<td>170</td>
</tr>
<tr>
<td>C.1</td>
<td>Evidence primitives used in ontology alignment by relation</td>
<td>184</td>
</tr>
</tbody>
</table>
List of Figures

1.1 Example of mapping classes cross-ontology using relationships within the original ontologies. .................................................. 8

2.1 Example fragment of the Gene Ontology (GO). ..................... 17

2.2 Example OWL fragment of the Gene Ontology (GO). ............. 19

2.3 Example alignment of fragments of the Gene Ontology (GO) and Mammalian Phenotype (MP) Ontology. ............................. 23

2.4 The Ontology Spectrum, by Obrst in [Daconta et al., 2003] ....... 25

2.5 Simplest ontology alignment process. .................................. 26

2.6 Possible patterns for output of the ontology alignment process. ... 27

3.1 Schema alignment with COMA [Do and Rahm, 2002] ............. 35

4.1 Example of part_of relationship in GO, expressed in OWL. ...... 50

5.1 Subclass Pattern 1, based on Synonymy. ............................. 59

5.2 Example of Subclass Pattern 1. ....................................... 60

5.3 Subclass Pattern 2, based on hyponymy. ............................. 60
5.4 Subclass Pattern 3, based on hypernymy cross-ontology. .......................... 61

5.5 Pattern 1 for Inferred Relations, based on synonymy cross-ontology. . 62

5.6 Inferred Relation Pattern 2, based on hyponymy. ................................. 63

5.7 Scatter plot to be used in alignment evaluation. ................................. 66

5.8 ROC scatterplot showing performance of Subclass Pattern 1 which
  relies on synonymy cross-ontology. .................................................. 69

5.9 DET scatterplot showing error rates of Subclass Pattern 1. ............... 70

5.10 ROC scatterplot showing performance of Subclass Pattern 2 which
  relies on hyponymy cross-ontology. .................................................. 72

5.11 DET scatterplot showing error rates of Subclass Pattern 2. ............... 73

5.12 ROC scatterplot showing performance of Subclass Pattern 3 which
  relies on hypernymy cross-ontology. .................................................. 75

5.13 DET scatterplot showing error rates of Subclass Pattern 3. ............... 76

5.14 Overall performance of patterns to align using subclass relations. ... 77

5.15 Alignment accuracy of superclass patterns. ...................................... 78

5.16 ROC scatterplot showing the overall performance of inferred relations
  patterns. .................................................................................................. 81
7.3 Example of mapping classes cross-ontology using relationships within
the original ontologies. .................................................. 130

7.4 Branch & Bound and Alignment Performance Comparison for up to
1000 Pairs. ................................................................. 136

7.5 Branch & Bound and Alignment Performance Comparison for up to
1000 Pairs, including time to select pairs to align. ................. 137

7.6 Comparison of large-scale alignment accuracy of Branch & Bound and
Optimized Alignment using match coverage, an estimate of recall. . 138

7.7 Comparison of large-scale alignment accuracy of Branch & Bound and
Optimized Alignment using match ratio, an estimate of precision. . 139

7.8 Comparison of match coverage performance of features to detect se-
semantic closeness. ....................................................... 140

7.9 Match coverage for alignment of large-scale biomedical ontologies, us-
ing subclass relationships. Average over 3 SVM models. ........ 141

7.10 Match ratios for alignment of large-scale biomedical ontologies, using
subclass relationships. Average over 3 SVM models. .............. 143

8.1 Mouse Genome Informatics (MGI) Search site. .................. 160

8.2 Results of gene query on MGI Search site. ....................... 161
### 8.3 Applying ontology alignment to enhance searches of biomedical ontologies.

8.3 Applying ontology alignment to enhance searches of biomedical ontologies.

8.4 Notional example of ontology alignment in the biomedical domain.

Knockout Mouse Project results could be enriched through aligned
gene and phenotype knowledge sources.

8.5 Application of alignment: Web Service Composition.

8.6 Application of alignment: Web Service Mediation.

8.7 Role based alignment could utilize user meta data to enhance the alignment process.

### A.1 Example reference alignment in the Academic domain.

### A.2 Example reference alignment of inferred relations in the Academic domain.

### A.3 Data representation of the reference alignment, semi-automatically generated.

### B.1 Example reference alignment in the Biomedical domain: Platelet Activation.

### B.2 Example reference alignment in the Biomedical domain: Bone Remodeling.
Chapter 1

Introduction

The goal of this research is to advance the state of the art in ontology alignment. We contribute in three primary areas. First, we extend the information obtained in alignment by aligning with relationships beyond similarity and equivalence; specifically, we align ontologies using subclass and superclass relations and infer alignments cross-ontology using relationships contained within the original ontologies. Second, we contribute original algorithms and techniques to align ontologies. We identify new evidence for ontology alignment by showing that the use of semantics in conjunction with upper ontologies and other linguistic resources can enhance the alignment process. We further enhance performance of this approach by modeling the ontology alignment problem in Support Vector Machines. Third, we apply our algorithms to align large-scale biomedical ontologies. We contribute a solution to scale ontology alignment with an original algorithm that bounds processing without significant reduction in precision.

In this Chapter, we provide background on ontologies and ontology alignment. We provide detail on our motivation for this work. We summarize our contributions and describe the organization of this dissertation.
1.1 Background

Inspired by [Neches et al., 1991], [Gruber, 1993], [Guarino and Giaretta, 1995] and [Borst., 1997], we define an ontology as *a partial, formal specification of a shared conceptualization that includes basic terms and relationships of a domain vocabulary as well as rules for combining the terms and relationships*. Ontology has its roots in a number of technologies, including formal ontology in philosophy, formal semantics in linguistics, knowledge representation in Artificial Intelligence (AI) and logic and formal methods in theoretical computer science. In some sense, an ontology can be viewed as a next step in the evolution of database systems, since an ontology supports explicit expression of semantics. This is an improvement over the capabilities of databases which provide for representation of only syntax and implicit semantics. Standards for representing ontologies started to emerge in the 1990s; for example, the Knowledge Interchange Format (KIF). Developed as part of the DARPA Knowledge Sharing Initiative\(^1\), KIF is a language that supports the expression and interchange of knowledge across heterogeneous systems using declarative semantics [Geneserth and Fikes, 1992]. KIF eventually became the basis of Common Logic\(^2\), a new ISO standard as of October 2007. More recently, the Web Ontology Language (OWL) [McGuinness and van Harmelen, 2004] has emerged as a recommended World Wide Web Consortium (W3C) standard.

With the advent of the Semantic Web [Berners-Lee et al., 2001], standard methods for defining the semantics of terms are allowing concept sharing on a wide scale.

---

2. http://cl.tamu.edu/
Ontologies represented in a standard language, such as OWL, and exposed on a network offer the potential for unprecedented interoperability solutions since they are semantically rich, machine-interpretable and inherently extensible. The number of ontologies available on the Web has increased significantly in recent years, as evidenced by searches of Swoogle\(^3\), Protege Ontology Library\(^4\) OntoLingua\(^5\) and the DAML Ontology Library\(^6\). In addition, there are large numbers of biological ontologies being utilized broadly; for example, the Open Biomedical Ontologies [Smith et al., 2004] and the Gene Ontology [Ashburner et al., 2000]. There has been success in applying ontologies for integration in biotechnology [Sahoo et al., 2007], [S. and Bodenreider, 2007], [Elmasri et al., 2007], [Tiffin et al., 2005] and in prototypes for interoperability in defense applications [Matheus, 2005], [Stoutenburg et al., 2007]. However, the potential power of ontological solutions is curtailed by the difficulty and expense of building and aligning ontologies manually. In addition, scalability in the development and alignment of large-scale ontologies remains a grand challenge [Hepp et al., 2008]. And, despite use of a standard language for ontology representation, such as OWL, there are still vast differences in ontology design. As different parties generate ontologies independently, the level of heterogeneity across platforms increases [Euzenat and Shvaiko, 2007].

To reduce the impediments to integrating data sources using ontologies, researchers are seeking to apply ontology alignment techniques to automatically dis-

\( ^3 \)http://swoogle.umbc.edu
\( ^4 \)http://protegwiki.stanford.edu/index.php/Protege Ontology Library
\( ^5 \)http://www-ksl-svc.stanford.edu:5915
\( ^6 \)http://www.daml.org/ontologies
cover relationships across ontological components. Ontology alignment (also referred to as matching [Euzenat and Shvaiko, 2007], [Rahm and Bernstein, 2001], mapping [Noy, 2004], and merging [Noy and Musen, 2003]), is the process of discovering relationships between entities across two or more ontologies based upon a variety of techniques that use entity labels, structure, semantics, and external resources to determine the relationships. Schema matching is also a related term that refers to earlier work in matching, primarily over databases [Doan et al., 2004]. The differences between schema integration and ontology alignment are well defined in [Noy, 2004]. We will use the term *ontology alignment* in this work. Ontology alignment is viewed as a potential solution to a broad set of challenges, including data integration, agent negotiation, and Web service composition [Ehrig, 2007] as well as peer-to-peer information sharing and advanced navigation and query answering on the Web [Euzenat and Shvaiko, 2007]. In addition, ontology alignment has been applied for rapid enterprise integration in government research projects [Stoutenburg et al., 2007].

There is a significant body of work in the area of schema matching and ontology alignment, with well over 70 approaches published to date. Nearly all of the approaches seek to learn similarity and equivalence relations between entities across schemas or ontologies; only a few have sought to learn others [Palopoli et al., 2003], [Kotis et al., 2006], and even those are restricted to hypernymy and hyponymy. Some solutions discover subsumption [Kim et al., 2003]. Within recent months, approaches have emerged for subclass relation alignment [Hamdi et al., 2008], [David, 2008] [Jean-Mary and Kabuka, 2008]. Therefore, acquiring additional properties between ontological components remains a research challenge [Zimmerman et al., 2006], [Euzenat and
Interestingly, at the recent Ontology Alignment Evaluation Initiative (OAEI)\(^7\), held in late 2008, it was observed that “there are now test cases where non equivalence alignments matter” [Caracciolo et al., 2008]. Therefore, it is very timely to propose approaches to alignment that use non equivalence relationships.

Many believe that alignment techniques will be enhanced with the use of upper ontologies but no system existed in 2007 that applied upper ontologies [Euzenat and Shvaiko, 2007]; as of the OAEI 2008, no such system is described [Caracciolo et al., 2008]. In OAEI 2006, only one application used reasoning techniques that exploited the semantics in OWL [Euzenat et al., 2006], though more applications have begun using that approach within the past 15 months [Euzenat et al., 2007], [Caracciolo et al., 2008]. Finally, despite the very large solution space, recent evaluation results show that there is still significant work to apply ontology alignment techniques to solve real world problems [Euzenat et al., 2006] and large-scale ontology alignment remains a serious challenge [Hepp et al., 2008]. This research focuses in these key areas.

1.2 Motivation

There are many compelling reasons to study and advance the state of the art in ontology alignment. Applications of ontology alignment include enterprise data integration, agent negotiation, peer-to-peer information sharing and advanced navigation and query answering on the Web [Euzenat and Shvaiko, 2007], [Ehrig, 2007]. In ad-

\(^7\)http://oaei.ontologymatching.org/2008/
dition, work is starting to emerge in which ontology alignment is applied for dynamic Web service composition and mediation [Hibner and Zielinski, 2007]. However, one of the most compelling use cases for ontology alignment is in the area of supporting translational research, which is the movement of discoveries in basic research to application at the clinical level [Ruttenberg et al., 2007]. There has been tremendous growth in the number and use of ontologies in biomedicine in recent years. Industry and government have made multi-million dollar investments in biomedical ontologies and funded numerous grants for knowledge representation in ontologies. Hundreds of rich biomedical ontologies have emerged in recent years. For example, the Gene Ontology (GO) [Ashburner et al., 2000] and Online Biomedical Ontologies (OBO) [Smith et al., 2004] have been developed in recent years. Now that these large bodies of biomedical knowledge exist, there is a need to better utilize and integrate this knowledge. Ontology alignment techniques can be applied to build bridges across knowledge silos in order to advance translational research [Ruttenberg et al., 2007] and generate new hypotheses in disease research and discover new knowledge, particularly cross-organism relationships. Furthermore, as stated in [Ruttenberg et al., 2007], a key goal of the National Institute of Health (NIH) Roadmap\(^8\) is to strengthen translational research in order to speed the translation of scientific discoveries into practical applications for the patient. The ability to integrate data across biomedical domains would enable knowledge discovery with potential clinical application [Ruttenberg et al., 2007]. One of the goals of this research is to identify, apply and develop ontology alignment techniques to integrate biomedical knowledge sources and work

\(^8\)http://nihroadmap.nih.gov/
toward developing a capability that will have utility in the area of translational research. Our vision for future uses of this technology, particularly in the biomedical domain, are described in Chapter 8.

1.3 Summary of Research Goals and Contributions

The goal of this work is to advance the state of the art in automated ontology alignment by contributing in three main areas. First, we contribute an approach to extend the information that can be derived in ontology alignment; specifically, we seek to discover non-equivalence relations across ontological classes. We present a set of algorithms to discover three such relation types, namely subclass, superclass and relations within the original ontologies that are used to align concepts across ontologies. The latter is particularly powerful since it allows generic relation discovery across ontologies. To illustrate this idea, we present an example in Figure 1.1. In this case, since it is given that (PlateletActivation, partOf, BloodCoagulation) in ontology$_{1}$, and (BloodCoagulation, synonymy, Clotting), it is inferred that (PlateletActivation, partOf, Clotting) is one likely alignment between ontology$_{1}$ and ontology$_{2}$. We discover (BloodCoagulation, synonymy, Clotting) through the use of external knowledge sources. This type of alignment is discussed further in Chapters 5 and 6.

Second, we contribute new methods and techniques to enhance the ontology alignment process. We show that the use of OWL semantics in conjunction with upper ontologies and online linguistic resources is effective in the alignment process and specifically contributes to the discovery of new properties. We model the ontology
alignment problem in a Support Vector Machine and show that ontology alignment is enhanced significantly by the use of an SVM.

Finally, we investigate approaches to scaling ontology alignment in order to align large-scale biomedical ontologies. Specifically, we discuss scalability issues in the area of processing, reasoning, and aligning very large ontologies. We survey related work in the area and discuss a number of approaches that we developed to improve scalability and performance in ontological processing and reasoning. We present an alignment algorithm that bounds the processing by selecting optimal subtrees to align. We show that our work improves the efficiency of alignment without significant reduction in precision. We also discuss some of the approaches we developed that were not feasible solutions, such as an algorithm that applies agglomerative clustering to partition the algorithm prior to alignment. We conclude with a vision of future research areas in ontology alignment.
1.4 Research Methodology

We approached this work by first selecting properties to use in the alignment process. We selected subclass and superclass, as well as a capability to infer relations cross-ontology, as described in the previous section and depicted in Figure 1.1. The latter, which we will call inferred relations, supports alignment using relationships contained within the original ontologies. Each of these relationships is formally defined in Section 2.4.

Next, we designed a set of patterns for alignment, using the semantics of OWL in conjunction with the upper ontology OpenCyc, one of the leading open source upper ontologies by Cycorp. We selected a simple domain for initial testing that is composed of class names that are primarily commonsense terms, the Academic domain. We selected six random ontologies from the Web in this domain, developed by different authors, and composed a reference alignment. A reference alignment, defined in [Euzenat and Shvaiko, 2007], is the set of “true” results. The reference alignment is used to evaluate the accuracy of our approach. Reference alignment development is a manually intensive task, requiring domain experts in some domains. We developed software utilities to aid in the process of developing reference alignments, so the alignments were developed semi-automatically. We tested the patterns and evaluated them against the reference alignment using precision and accuracy measures. The characteristics of classes and relations in the Academic domain are analyzed in Chapter 4 and the patterns and their performance are discussed in Chapter 5.

9http://www.cyc.com/cyc/opencyc/overview
To improve performance and agility in alignment, we modeled the ontology alignment problem in a Support Vector Machine. We developed feature extractors for each of the three relationship types and added many new features to the learning process. We tested the patterns in the Academic domain using SVMs and found that SVM technology significantly improved accuracy. Once we determined that SVM enhanced accuracy, we shifted our attention to alignment in the Biomedical domain. We collaborated with biologists\textsuperscript{10} to select ontologies to align. We decided to evaluate our approach with SVMs by aligning portions of the Gene Ontology (GO) [Ashburner et al., 2000] and the Mammalian Phenotype ontology, part of the Open Biomedical Ontologies (OBO) [Smith et al., 2004]. A reference alignment was developed with the expertise of the biological domain experts. We aligned components, measured the results and found that the approach was promising.

Next, we applied our SVM-based alignment approach to align large-scale biomedical ontologies. We encountered significant problems processing the large ontologies. Therefore, we decided to investigate scalability and emerging approaches to manage and align large-scale ontologies. We surveyed the state of the art and investigated an algorithm for partitioning large-scale ontologies that builds on agglomerative clustering. We developed an algorithm and unique cohesiveness measure that had the potential to improve the state of the art by considering the qualitative properties of the relations in ontologies. However, testing revealed that partitioning merely pushes the complexity of alignment to an a priori process. In fact, preliminary results sug-

\textsuperscript{10}Our sincere thanks to Dr. Lisa Hines and graduate student Kaily Ewing for their invaluable expertise.
gested that overall processing time would be increased. Therefore, we investigated other approaches and developed a Branch and Bound algorithm to prune unlikely alignment candidates. We showed that this approach improved alignment efficiency without significant reduction in precision, an important contribution to improving overall scalability in alignment. Once we could process and align large-scale ontologies, we performed further ontology alignment experiments in the biomedical domain. These results are presented in Section 7.5.

During this work, we saw many opportunities for future research. In particular, we have ideas on how to apply ontology alignment for enhanced data analysis techniques and hypothesis generation in the biomedical domain. Ontology alignment could be applied for enterprise integration and dynamic Web service mediation and composition. We also have ideas for role-based ontology alignment. These ideas for future research are presented in Chapter 8.3.

1.5 Organization of Dissertation

This dissertation is organized as follows. First, we provide a set of definitions in Chapter 2. We present a formal definition of ontology and discuss issues with the representation of ontologies. We also present a formal definition of ontology alignment and define the relationships used in this work. In Chapter 3, we discuss related work. We discuss the different types of alignment approaches, including alignment techniques over schemas, instances and ontologies, as well as hybrid approaches. We survey the best performing approaches and discuss the use of machine learning in
alignment. We also highlight the gaps in capability in this chapter and discuss how this work is designed to address those gaps. In Chapter 4, we describe the domains of ontologies used in this work, including the Academic and Biomedical domains. We examine the characteristics of ontologies in these domains and discuss implications in alignment. Next, in Chapter 5, we present unique patterns for alignment using the relationships of subclass, superclass and inferred relations. We evaluate these patterns using ontologies from the Academic domain. We discuss sources of error and how those issues might be addressed. In Chapter 6, we provide a brief description of Support Vector Machine technology and show how the ontology alignment problem can be modeled in a Support Vector Machine. We evaluate the approach using ontologies from the Academic domain and demonstrate that ontology alignment accuracy is significantly enhanced using Support Vector Machines. Once we establish that SVMs improve alignment performance, we apply our approach to align ontologies in the Biomedical domain. We evaluate the approach and present an architecture for ontology alignment. Finally, in Chapter 7, we describe the challenges in scaling ontology processing, reasoning and alignment. We survey emerging approaches to address the challenges. We describe the work we performed to manage large ontologies and present a novel algorithm that improves alignment efficiency without significant reduction in accuracy. We also discuss some of the infeasible approaches to enhancing alignment.
Chapter 2
Definitions

In this chapter, we provide formal definitions for ontology and ontology alignment. We also discuss issues with the representation of ontologies and define precisely the relationships used to align ontologies in this work.

2.1 Ontology

An early definition of ontology in the context of computer science was given by Neches [Neches et al., 1991]: *An ontology defines the basic terms and relationships comprising the vocabulary of a topic area as well as the rules for combining terms and relations to define extensions to the vocabulary.* This is an important definition as it provided for axioms to be part of ontology. Gruber defined an ontology as *an explicit specification of a conceptualization* [Gruber, 1993]. Guarino and Giaretta found it necessary to weaken Gruber’s definition to be a *partial account of a conceptualization* in their discussion of (at the time) emerging definitions of ontology and related terms in knowledge engineering [Guarino and Giaretta, 1995]. The weakening of the definition is necessary because Gruber’s notion of conceptualization is extensional while Guarino and Giaretta argue it should be intensional [Obrst, 2006]. Further, Guarino and Giaretta define Ontology (large O) as *that branch of philosophy which deals with*
the nature and the organization of reality. Also, they clarify the multiple definitions and uses of ontology (small o), the final result being two senses; sense 1 being a *logical theory which gives an explicit, partial account of a conceptualization*; and sense 2 being a *synonym of conceptualization*. They define conceptualization as an *intensional semantic structure which encodes the implicit rules constraining the structure of a piece of reality* [Guarino and Giaretta, 1995]. Borst proposed that ontologies are defined as a *formal specification of a shared conceptualization* [Borst, 1997]. The idea that ontologies provide explicit and formal specification became important as proposed standard ontology languages began to emerge, such as The Web Ontology Language (OWL) [McGuinness and van Harmelen, 2004]. Explicit definition also supports the ability of software to operate over ontologies in a semantically rich way. Therefore we prefer to define ontology as a *partial, formal specification of a shared conceptualization that includes basic terms and relationships of a domain vocabulary as well as rules for combining the terms and relationships*.

There are many formal definitions of ontology that have been published. Typically, definitions are designed to meet the needs and goals of the researcher but there are some core components that comprise any definition. At a minimum, an ontology can be defined to consist of a nonempty set of classes, a nonempty set of properties or relations, a hierarchy among classes, a hierarchy among properties, and a set of functions that map classes via properties, the latter being an expression of class relationships. Ehrig defines this set of features as *Core Ontology* [Ehrig, 2007]. Some definitions of ontology include instances [Euzenat and Shvaiko, 2007] while others define instances and the instantiation function to be part of a *Knowledge Base* [Ehrig,
Interestingly, Euzenat explicitly defines a subset of relations that exist in an ontology, such as subclass (he calls it specialization), disjointness (exclusion), instantiation and assignment [Euzenat and Shvaiko, 2007]. Euzenat also defines data types to be part of an ontology while Ehrig defines data types to be a special case of class. Also, Ehrig states that an ontology must consist of a lexicon that defines the names of classes and relations and a lexical reference for those names.

We prefer Ehrig’s approach to defining ontology since it provides a modular definition of the components of an ontology in such a way that operations and algorithms can be more precisely defined. However, we use some elements of both approaches in the definition below.

Definition 1 Ontology

An ontology is a 6-tuple $O = < C, R, \leq_{C}, \leq_{R}, \sigma, A >$, such that:

- $C$ is a nonempty set of classes: $\{C_1, C_2, \ldots C_n\}$,
- $R$ is a nonempty set of relations: $\{R_1, R_2, \ldots R_n\}$,
- $C, R$ are disjoint,
- $\leq_{C}$ is a class hierarchy, a partial order on $C$,
- $\leq_{R}$, is a relationship hierarchy, a partial order on $R$,
- $\sigma : R \rightarrow C \times C$, representing relationships between classes,
- $A$ is a set of class axioms, possibly empty: $\{A_1, A_2, \ldots A_n\}$.

Note that in this work, we do not plan to operate over instance data. Therefore, we
leave instances out of the definition of ontology. Also, the term *property* should be considered synonymous with *relation* and *relationship*.

Similar to Ehrig’s definition, we denote that if \( c_1 <_C c_2 \), where \( c_1, c_2 \in C \), then \( c_1 \) is a subclass of \( c_2 \) and \( c_2 \) is a superclass of \( c_1 \). If \( c_1 <_C c_2 \), and \( \nexists \ c_3 \in C \) such that \( c_1 <_C c_3 <_C c_2 \), then \( c_1 \) is a direct subclass of \( c_2 \) and \( c_2 \) is a direct superclass of \( c_1 \).

We also borrow, in part from Ehrig, the concept of a set of *entities* and we say that an entity \( e \in E \) in ontology \( O \) is a class or relation, such that \( e \mid O \in C \cup R \). We treat data types as a set \( D \subseteq C \), i.e., a special set of classes, as in [Euzenat and Shvaiko, 2007].

To illustrate these concepts, we present an example of an ontology from the Biomedical domain in Figure 2.1. This example was extracted from the Gene Ontology [Ashburner et al., 2000]. In this example, concepts, shown in oval shapes, are related via semantic relationships represented by directed links. The concept *PlateletActivation* is defined to be a subclass of *CellActivation* and also is partOf *BloodCoagulation*. The concept *RegulationOfPlateletActivation* regulates *PlateletActivation* and is defined to have two subclasses, *NegativeRegulationOfPlateletActivation* and *PositiveRegulationOfPlateletActivation*. In addition, *NegativeRegulationOfPlateletActivation* negativelyRegulates *PlateletActivation* and *PositiveRegulationOfPlateletActivation* positivelyRegulates *PlateletActivation*. As can be seen in this figure, an ontology is typically represented as a directed acyclic graph.
2.2 Representation of Ontologies

Representation of ontologies is an important consideration in ontological engineering and ontology alignment. This is because the representation language can greatly enhance or limit the expressiveness of an ontology, and therefore its effectiveness.

Standards for representing ontologies started to emerge in the early 1990s. Early standards were based on first order logic, frames combined with first order logic and description logics [Gomez-Perez et al., 2004]. For example, CycL [Lenat and Guha, 1990] was one of the first ontology languages to be developed, based on frames and first order logic. The Knowledge Interchange Format (KIF) language was developed as part of the DARPA Knowledge Sharing Initiative [Geneserth and Fikes, 1992]. KIF is a language that supports the expression and interchange of knowledge across heterogeneous systems using declarative semantics [Geneserth and Fikes, 1992]. KIF
eventually became the basis of Common Logic, a new ISO standard as of October 2007. In the late 1990s, as the Internet began to emerge, mark-up languages were developed to represent ontologies. XML-based languages emerged, including RDF\(^1\) and RDF Schema\(^2\), which were developed by the World Wide Web Consortium (W3C). RDF provides a basic capability that allows concepts (or "resources") to be connected in a directed graph using relationships. Resources are accessed via a Uniform Resource Locator (URL), supporting the notion of distributed Web resources. RDF Schema enriches RDF by providing a vocabulary for describing properties and classes of RDF resources, with a semantics for specifying generalization-hierarchies of such properties and classes.

The Web Ontology Language (OWL) [McGuinness and van Harmelen, 2004] is a language developed to express ontologies for the Semantic Web and is the most expressive of ontology languages being defined for the Web [Daconta et al., 2003]. OWL builds on RDF and RDF Schema by providing additional expressivity for describing properties and classes, including specific relations between classes (e.g. union, equivalence, disjointness), cardinality (e.g. "exactly one"), equivalence, characteristics of properties (e.g. symmetry), and enumerated classes. OWL is defined in three sub-languages, including OWL Full, OWL Description Logic (DL), and OWL Lite, listed in order of decreasing expressivity. OWL Full includes all OWL and RDF constructs and allows classes to be treated as individuals, which enables meta-modeling. For example, OWL Full allows specification of *FordBronco* as a class of cars and also as

\(\text{http://www.w3.org/TR/2004/REC-rdf-syntax-grammar-20040210/}\)

\(\text{http://www.w3.org/TR/rdf-schema/}\)
Figure 2.2: Example OWL fragment of the Gene Ontology (GO).

OWL Full is not decidable, and thus few implementations of OWL Full exist. OWL DL includes most OWL language constructs, including \textit{disjointWith}, \textit{oneOf} and \textit{cardinality}, among others, which are not included in OWL Lite. OWL DL is considered to be computationally complete, decidable, and tractable.

OWL Lite is restricted to a subset of the OWL DL constructs. OWL became a W3C recommendation in February 2004.

For this work, we selected ontologies that are expressed in OWL; most ontologies use OWL Lite, though our work could also operate over ontologies in RDFS and OWL DL. We selected ontologies in OWL because OWL is the W3C standard; and a recent study estimates that 82% of ontologies on the Web are in RDFS or OWL [Hu et al., 2008]. An example extract of an ontology expressed in OWL used in our work is shown in Figure 2.2. This is an excerpt of the same class from the ontology shown in Figure 2.1, \textit{PlateletActivation}.
2.3 Ontology Alignment

*Ontology alignment,* also referred to as *ontology matching,* is the process of identifying relationships between entities across ontologies. Noy and Musen defined ontology alignment as *establishing different kinds of mappings (or links) between two ontologies, thus preserving the original ontologies* [Noy and Musen, 1999]. Klein [Klein, 2001] and Ehrig [Ehrig, 2007] define ontology alignment to be a process in which, for each entity in a given ontology, a corresponding entity in a second ontology is identified that has the same intended meaning. Therefore, they constrain the relationship between entities across ontologies to be one-to-one equivalence. We extend that definition by allowing any type of relationship across entities. We also restrict our definition such that instances are not considered as entities.

To formally define ontology mapping, we first define correspondence, inspired by [Ehrig, 2007]. This definition uses the same sets $C$, $R$, and $E$ defined above.

**Definition 2 Correspondence**

*Consider two ontologies $o_i$, $o_j$. Let $C_i$ and $R_i$ represent classes and relations in $o_i$, respectively. Let $C_j$ and $R_j$ represent classes and relations in $o_j$, respectively. A correspondence $CORR$ is a 4-tuple $= < e_i, e_j, \phi, p >$, where:*

- $e_i \in o_i, e_j \in o_j$,
- if $e_i \in C_i$, then $e_j \in C_j$, or if $e_i \in R_i$ then $e_j \in R_j$,
- $\phi \in$ a nonempty set of relationships (not necessarily $\in R_i \cup R_j$), and,
- $p \in \mathbb{R}$ such that $p \in [0, 1]$, denoting a confidence on the relationship $\phi$. 


We define $\phi$ as the set of relationships that will be acquired using ontology alignment techniques. These relations may or may not exist in the ontologies. For example, most ontology alignment techniques seek to acquire equivalence across ontological entities. In this work, we seek to acquire non-equivalence relations. Our work is also distinguished from previous work in ontology alignment because we seek to discover what we call *inferred relations*; that is, we seek to acquire relations between ontological entities $e_i, e_j$ that exist in at least one of the ontologies. Formally, we seek to acquire relations $r | r \in R_i \cup R_j$. This is shown in Figure 1.1 and explained further in the next Section.

Given this definition of correspondence, we can now define an ontology alignment, which is very similar to [Ehrig, 2007].

**Definition 3 Ontology Alignment**

*An ontology alignment is a set of correspondences.*

Note that this is a weak definition of ontology alignment; this is intended, because in this work, all entities in each ontology need not necessarily be in the set of discovered correspondences. In this work, the term *alignment* should be considered synonymous with correspondence.

To decide if a particular correspondence exists, we will apply multiple approaches, including string-, structure- and other classic techniques, as well as original algorithms that use the semantics of OWL and upper ontologies. (This is discussed in further detail in Chapters 5 and 6.) In this work, we seek to discover correspondences that comprise new properties between ontological entities beyond equivalence and similarity. To acquire new relationships, evidence for each is specified. This evidence
is essentially a set of features, the significance of which is learned by a Support Vector Machine. Each relation will be acquired using a different set of evidence.

To illustrate the idea of ontology alignment, we present an example in Figure 2.3. In this example, we are given two ontologies. The first ontology is a subset of the ontology previously seen in Figures 2.1 and 2.2. The second ontology is an excerpt from the Mammalian Phenotype ontology [Smith et al., 2004]. In this example, a number of alignments could be acquired cross-ontology. The example shows the alignments that should be identified between AbnormalPlateletActivation in MP and multiple related concepts in GO. First, we expect that AbnormalPlateletActivation should be identified to be a subclass of PlateletActivation. From that relation, using the patterns described in Chapter 5 and other linguistic and semantic features, it should be discovered that AbnormalPlateletActivation is also a subclass of CellActivation and that AbnormalPlateletActivation is partOf BloodCoagulation. In addition, we would expect to discover that RegulationOfPlateletActivation regulates AbnormalPlateletActivation, since the latter is a subclass of PlateletActivation. The same is true for NegativeRegulationOfPlateletActivation and PositiveRegulationOfPlateletActivation.

A more compelling example that illustrates the potential power of this technology is the use of ontology alignment techniques to integrate knowledge from biomedical experiments. This and other uses of ontology alignment are described in further detail in Section 8.3.
Figure 2.3: Example alignment of fragments of the Gene Ontology (GO) and Mammalian Phenotype (MP) Ontology.

2.4 Relationships in Ontology Alignment

One of the primary goals of this research is to acquire relationships between ontological entities beyond similarity and equivalence. These relationships include the following.

- **Subclass**

- **Superclass**

- Correspondences such that $\phi \in R_i \cup R_j$; that is, we will discover properties across ontological entities that are defined in the original ontologies, as depicted in Figure 1.1. We will define these set of relations to be *inferred relations* across ontologies.

23
To define these relations further, we first distinguish between term and concept. A term (could be a word or a phrase) indexes a concept, while a concept is the meaning of the term [Obrst, 2006]. There is not necessarily a one-to-one relation between a term and a concept; it may take one or more concepts to express the meaning of a term, and a concept may be indexed by more than one term [Obrst, 2006].

Obrst also defines “The Ontology Spectrum,” in which clear distinction is made between the weak subclassificationOf relation and the strong isSubclassOf relation [Daconta et al., 2003]. The Ontology Spectrum is presented in Figure 2.4. This continuum shows that the relation subclassificationOf is a weaker notion, typically associated with weak taxonomies. The isSubclassOf relation is transitive, associated with the strongest semantics, a conceptual model or logical theory. The relations broader-than and narrower-than indicate generalization and specialization in a broad sense and are associated with strong taxonomies, but are term-based. OpenCyc uses broader-than and narrower-than in its concept hierarchies. These are synonymous with the relations hypernymy and hyponymy, used in WordNet. Subsumption is a relation that is essentially a partial ordering on a set. Subsumption can be considered to be synonymous with subclass. In this work, we seek to align using the strongest subclass relation, and its symmetric relation, superclass.

According to [Obrst, 2006], a “weak” taxonomy is characterized by the use of subclassificationOf relations to link terms and parent-child relations may not be well defined or are not consistently applied throughout the taxonomy. A “strong” taxonomy, on the other hand, is defined as one of two types. The first type is a strong term taxonomy; for example, a thesaurus that uses narrower-than. The second type
is a strong concept taxonomy, which can be a conceptual model or logical theory that uses subclass relations.

### 2.5 The Process of Ontology Alignment

In this section, we give a brief, informal discussion of the ontology alignment process. Ontology alignment typically considers two input ontologies. An alignment process is applied, which typically consists of string, linguistic, structure, and semantic based approaches. The result is a set of data, often in XML format, consisting of the set of correspondences. Each correspondence consists of a class pair, the relation that is shared and the confidence in the existence of the relationship. This simple process is shown in Figure 2.5. In this work, we predict discrete alignment, so the confidence value is in the set \([-1, 1]\). Many additional components may be added to this process,
such as a feedback loop for user interaction; input from external knowledge sources; and combination rules used at the latter parts of the process.

In addition, there are multiple patterns for alignment results. Output of an alignment process may be a new ontology that contains the new relationships between entities, or it could be modifications to the existing ontologies to indicate the new relationships. The alignment process could also result in a set of correspondences that could be used as a resource for autonomous communication systems, peer to peer systems or dynamic Web service composition and mediation. These patterns are shown in Figure 2.6. In this work, we generate alignment results in XML using the pattern shown in Figure 2.6 (a).

In the next chapter, we discuss previous work and the state of the art in ontology alignment.
Figure 2.6: Possible patterns for output of the ontology alignment process.
Chapter 3
State of the Art in Ontology Alignment

3.1 Introduction

There is a significant body of work in the area of schema matching and ontology alignment, with well over 70 approaches published to date. A number of very good surveys have been published in recent years that document the features of the various approaches [Wache et al., 2001], [Rahm and Bernstein, 2001], [Euzenat et al., 2004], [Kalfoglou and Schorlemmer, 2003], [de Bruijn et al., 2006], [Euzenat and Shvaiko, 2007]. Ehrig [Ehrig, 2007] also provides a description of some of the leading techniques. Doan and Halevy [Doan and Halevy, 2005] provide a comprehensive review of matching techniques from a database perspective. Noy provides an excellent discussion of the differences and similarities between database schema matching and ontology alignment [Noy, 2004].

None of these surveys provide specific comparisons of accuracy and performance. This is likely due to the fact that wide agreement on metrics for ontology alignment comparison have only started to emerge [Euzenat and Shvaiko, 2007]. In fact, evaluations of matching systems remains a future research challenge [Euzenat and Shvaiko, 2007], [Ehrig, 2007]; in particular, there is a need for high quality data sets that capture different types of alignment challenges, definition of application-specific settings
to support comparisons, and evaluation measures for different types of tasks [Euizenat and Shvaiko, 2007]. New methods for calculating precision and recall are being proposed to distinguish finer degrees of accuracy [Ehrig and Euizenat, 2005], [Kirsten et al., 2007] and replacements for long standing traditional matching methods, such as string comparisons, are also being introduced, tailored for the ontology alignment problem [Stoilos et al., 2005]. Despite the challenges, some evaluation initiatives have been making progress in this area, such as the Ontology Alignment Evaluation Initiative, discussed in the previous chapter. More details on evaluation of ontology alignment approaches are discussed in Section 8.2.2.

A broad set of approaches to alignment are summarized in [Stoutenburg, 2007], with specific details for each, including input, output, matching techniques and use cases. We summarize the notable characteristics and approaches here. The classification we use is inspired in part by [Euizenat and Shvaiko, 2007], except that we distinguish between alignment over schemas using structure vs. alignment over ontologies using structure.

### 3.2 Alignment Techniques over Schemas

Early solutions to the alignment problem operated over database schemas primarily. Some use string-based techniques exclusively, such as the Data Element Tool-based Analysis (DELTA) [Clifton et al., 1997], though most use a variety of structure-based techniques. DELTA focused on document similarity, while others focused on data transformation as a goal [Milo and Zohar, 1998]. The Database
Intensional Knowledge Extractor (DIKE) tool constructs a mediated schema from heterogeneous data sources and explores the discovery of relationships other than similarity, specifically, hypernymy and hyponymy [Palopoli et al., 1998]. In addition, DIKE supports Cooperative Information Systems (CISs) [Palopoli et al., 1998]. One notable approach offers an extensible framework to integrate different matching algorithms, called COMbination of MAtching Algorithms (COMA) [Do and Rahm, 2002]. COMA remains one of the top performing approaches and was later enhanced to operate over ontologies and instances, as COMA ++ [Engmann and Massmann, 2007].

Some tools support schema evolution. The Type Evolution Software System (Tess) supports schema evolution by aligning components between versions and supporting transformation to the new format [Lerner, 2000]. The Toronto Mapping Adaptation System (ToMAS) automatically detects and corrects out-dated mappings [Velegrakis et al., 2003].

Cupid [Madhavan et al., 2000] applies domain specific thesauri and linguistic sources along with a broad set of matching algorithms to compute similarity. Melnik et al. use similarity propagation over a graph to compute similarity [Melnik et al., 2002]. Xu and Embley [Xu and Embley, 2003] also propose an alignment approach that operates over graphs. Others base the alignment strategy on clustering, such as Artemis [Castano et al., 2006a] and XClust [Lee et al., 2002]. MetaQuerier [Chang et al., 2005] supports querying over Web databases. Techniques that operate over database schemas continue to emerge, as in [Mork et al., 2006] which seeks to move schema alignment into mainstream operational use.
3.3 Alignment Techniques over Instances

A number of approaches focus on similarity computation using instance data. T-tree by Euzenat, was one of the first, and generates classes and class hierarchies (taxonomies) from instance data using bridge inference techniques [Euzenat, 1994]. Lacher and Groh use instance data for semi-automatic alignment to support document exchange [Lacher and Groh, 2001]. Formal Concept Analysis (FCA)-Merge applies formal concept analysis to merge ontologies that share instances [Stumme and Madche, 2001]. Doan et al. propose the Learning Source Descriptions (LSD) approach which discovers patterns and rules in training sets that consist of manually-developed schemas [Doan et al., 2001]. Doan et al. later enhanced LSD, proposing GLUE, which also applies learning techniques to discover semantic mappings between classes [Doan et al., 2004]. iMAP [Dhamankar et al., 2004] is notable because it was one of the first to support discovery of complex mappings between elements; i.e., mappings that require functions in order to properly align entities. iMAP also uses domain knowledge sources to enhance alignment discovery. Automatch, proposed by Berlin and Motro discovers attribute mappings between schemas by acquiring probabilistic information from manually-matched schemas and applying that information to compute matches [Berlin and Motro, 2002]. Wang et al. were among the first to use instance data to discover alignments between Web databases [Wang et al., 2004]. The approach entitled sPL Map [Nottelmann and Straccia, 2005] supports uncertainty in alignment by combining probability and logic theory. Other approaches based on instances include the statistical-based approach of Ichise et al. [Ichise et al., 2003], the approach of
Kang and Naughton [Kang and Naughton, 2005] which focuses on discovering relationships between attributes with "opaque" names, and [Bilke and Naumann, 2005], which identifies attribute alignment based on duplication among instances.

3.4 Alignment Techniques over Ontologies

Early work to align ontologies includes the work by [Hovy, 1998], which applies heuristics to match ontological elements and combine ontologies into a single reference ontology. The Semantic Knowledge Articulation Tool (SKAT) applies rules to semi-automatically discover mappings [Mitra et al., 1999]; it was succeeded by the ONtology compositiON system (ONION) [Mitra et al., 2000] which seeks to deliver a unified querying system over ontologies.

The PROMPT family of applications from Stanford is a set of tools for ontology alignment, merging and versioning. These tools include SMART [Noy and Musen, 1999], a linguistic-based algorithm to align ontologies, PROMPT [Noy and Musen, 2000], a tool for ontology merging, and PROMPT DIFF [Noy and Musen, 2002] which compares ontology versions. Anchor PROMPT [Noy and Musen, 2001] extends PROMPT to align ontologies, based on structure similarity measures. All of these capabilities are integrated into the Protege tool\(^1\).

Chimaera [McGuinness et al., 2000] is an ontology merging tool similar to PROMPT that is part of Ontolingua\(^2\). OntoBuilder learns an initial ontology based on a Web site and then merges it with related ontologies [Modica et al., 2001]. MapOnto

---

\(^1\)http://protege.stanford.edu

\(^2\)http://ksl.stanford.edu/software/ontolingua
[An et al., 2005] constructs complex mappings between schemas and ontologies. OntoMerge [Dou et al., 2005] supports ontology translation on the Web. H-Match [Castano et al., 2006b] represents an extension to Artemis, and automatically discovers alignments between ontological components in OWL. ASCO computes alignments between classes, relationships and between classes and relationships [Bach et al., 2004].

A small number of approaches have sought to learn relationships beyond similarity and equivalence. For example, HCONe relies upon heavy user interaction to compute alignments, and produces hypernymy, hyponymy and subsumption relations [Kotis et al., 2006]. MoA is an ontology merging and alignment tool that exploits linguistic information to acquire equivalence and subsumption [Kim et al., 2003]. Interestingly, CtxMatch treats the ontology alignment problem as a logical validity problem and applies description logic reasoners to compute alignment [Bouquet et al., 2003]; this approach allows computation of subsumption relationships between classes. S-Match was an enhancement to CtxMatch [Giunchiglia and Shvaiko, 2003].

Approaches that operate over the W3C standard OWL began to emerge even before OWL became a W3C recommendation in 2004, including [Noy and Musen, 2002] and [Bouquet et al., 2003]. Since then, most approaches that operate over ontologies operate over OWL.

### 3.5 Hybrid Approaches

Early approaches that use a combination of schema- and instance-based techniques include SEMINT [Li and Clifton, 1994] which aligns attributes across schemas
and Clio [Miller et al., 2000], which facilitates data transformation and integration of heterogeneous relational and XML schemas. SEMINT was one of the first approaches to use machine learning techniques to learn similarity formulas.

Naive Ontology Mapping (NOM) [Ehrig and Sure, 2004] builds upon the idea in COMA to compose matches and thus allows a wide variety of matching techniques to be used in the alignment computation. The efficiency of this system is improved with the Quick Ontology Mapping tool (QOM) [Ehrig and Sure, 2004]. oMap [Straccia and Troncy, 2005] also uses several matchers in the alignment computation.

Information-Flow-based Map, or IF-Map [Kalfoglou and Schorlemmer, 2003] matches ontologies by determining relationships to a common reference ontology. The OWL Light Aligner (OLA) [Euzenat and Valtchev, 2004] considers and balances the contribution of different components of an ontology to compute similarity. Alignment is computed based on string, language, and structure and in some cases, instances. It is worthwhile to mention the Wise-Integrator tool by [He et al., 2004] which integrates Web interfaces of search engines. Falcon-AO matches OWL ontologies using linguistic and graph matchers [Jian et al., 2005]. The Risk Minimization based Ontology Mapping (RiMOM) tool formalizes ontology alignment as a decision-making problem and is able to discover complex alignments [Tang et al., 2006]. Along with COMA, two other approaches, Falcon-AO and RiMOM, are among the best performing algorithms in OAEI 2006 and therefore, we discuss these approaches in greater detail in the next section. In addition, we highlight the best performers in OAEI 2007 and 2008.
3.6 Best Performing Alignment Approaches

Best performing approaches in recent years include COMA, Falcon-AO and Ri-MOM, as reported in the results of OAEI 2006\(^3\). Interestingly, each shares a similar architecture and approach. COMA (COMbining Matching Algorithms) is a schema-mapping approach that provides an extensible library of matching algorithms that can be combined in different ways. Users can add matching algorithms and modify combination strategies. COMA supports iterative matching with optional user feedback and allows reuse of previously generated alignments. The overall alignment approach used in COMA is re-printed in Figure 3.1 [Do and Rahm, 2002].

Experimental results determined that the default combination strategy put forth by COMA performed well [Do and Rahm, 2002], [Euzenat et al., 2006]. Averaging the values of all matching algorithms provided the best aggregation method. COMA was extended to match instances and ontologies in COMA++ [Engmann and Massmann, 2007].

[Falcon-AO aligns ontologies by using two approaches in parallel, a linguistic

\(^3\)http://oaei.ontologymatching.org/2006/results/
matcher and graphic matcher which exploits the structure. Falcon-AO then combines the independent results using a set of simple rules. The major strength of Falcon-AO lies in its ability to align ontologies that have little lexical similarity but high structural comparability, or which have strong lexical similarity. However, Falcon-AO performs less well when the structures between ontologies are very different; also, Falcon-AO experienced difficulty processing large ontologies [Jian et al., 2005]. Falcon-AO is similar to COMA in that is combines independent matcher results using rules.

The RiMOM (Risk Mitigation-based Ontology Mapping) technique is very similar to COMA and Falcon-AO. The main difference is that RiMOM is an approach that formalizes the mapping process into a decision making problem. Like COMA, RiMOM applies a multi-strategy execution approach in which mapping is performed independently then combined. Additionally, user interaction is optional and iteration is performed [Tang et al., 2006].

Since OAEI 2006, two other approaches have emerged as top performers in OAEI 2007 and 2008. These include Lily [Wang and Xu, 2007], [Wang and Xu, 2008a], and ASMOV [Jean-Mary and Kabuka, 2007], [Jean-Mary and Kabuka, 2008]. Lily is designed to solve issues in aligning heterogeneous ontologies. It uses a matching process that consists of extraction of semantic subgraphs, computation of similarity within subgraphs, and propagation of that similarity propagation across subgraphs [Wang and Xu, 2007]. In 2008, Lily was improved to use hybrid matching approaches, including multiple functions for generic matching, large-scale matching, semantic matching and ontology debugging4 [Wang and Xu, 2008a]. Developed in 2007, ASMOV is

---

4Ontology debugging is a method to detect some types of mapping errors [Wang and Xu, 2008b].
an alignment approach that measures of similarity over four feature types: lexical, internal structure (property restrictions), external structure (parents and children), and entity similarity. Each feature area is weighted. ASMOV also supports optional feedback from the user. Note that RIMOM remains a top performer through OAEI 2008.

In late 2008, three approaches were named in OAEI 2008 as aligning with non equivalence relations. These approaches include Aroma, Taxomap and ASMOV [Caracciolo et al., 2008]. AROMA aligns with equivalence and subsumption [David, 2008]. AROMA discovers subsumption by using mining association rules between data sources in conjunction with a statistical interestingness measure [David, 2008]. TaxoMap aligns with equivalence and subclass relationships, as well as a weak equivalence relation called isClose, to represent semantically related concept alignments [Hamdi et al., 2008]. The literature describing the work performed to align using non equivalence relations is fairly limited for ASMOV; the authors describe their work to align with equivalence in OAEI 2008 primarily [Jean-Mary and Kabuka, 2008].

### 3.7 Machine Learning for Alignment

A variety of machine learning techniques have been applied to schema and ontology alignment in the past. The first alignment approach to apply machine learning was SEMINT [Li and Clifton, 1994] which aligns attributes across schemas using neural networks. Clio [Miller et al., 2000], applies a type of naive Bayes classifier to facilitate data transformation and integration of heterogeneous relational and XML
schemas. Doan [Doan et al., 2001] proposed the Learning Source Descriptions (LSD) approach which used multiple learning techniques to discover patterns and rules in training sets that consist of manually-developed schemas. Doan et al. later enhanced LSD, proposing GLUE, which also applies learning techniques to discover semantic mappings between classes [Doan et al., 2004]. Automatch [Berlin and Motro, 2002] uses Bayesian learning to discover attribute mappings between schemas. This is accomplished by acquiring probabilistic information from manually-matched schemas and applying that information to compute matches. Xu and Embley [Xu and Embley, 2003] propose an alignment approach that operates over graphs using decision tree learning.

iMAP [Dhamankar et al., 2004] uses naive Bayes to support discovery of mappings between elements that require functions in order to properly align entities. BayesOWL converts ontologies into Bayesian networks and aligns ontological components by learning joint probabilities based on Web data [Pan et al., 2005]. The Ontology Mapping Enhancer [Mitra et al., 2005] also computes ontological alignment based on a Bayesian network. Based on Bayesian decision theory, the Risk Minimization based Ontology Mapping (RiMOM) tool formalizes ontology alignment as a decision-making problem [Tang et al., 2006].

The Support Vector Machine (SVM), first defined by Vapnik in [Vapnik, 1998], is a machine learning tool that supports classification, regression and outlier detection through supervised learning. SVMs have been applied successfully to real world problems such as face identification, intrusion detection, handwriting identification and more [Habib, 2008]. However, to our knowledge, Support Vector Machines have
not been used in ontology alignment.

3.8 Gaps in Capability

Despite the very large solution space, recent evaluation results show that there is still significant work to do to apply ontology alignment techniques to solve real world problems [Euzenat et al., 2006], [Euzenat et al., 2007], [Caracciolo et al., 2008]. In fact, in OAEI 2006, only one approach used reasoning to operate over the semantics of the models and none of the approaches exploited the characteristics of OWL to reason over the semantics of the ontological entities [Euzenat et al., 2006]. In the past 18 months, applications have emerged that are utilizing OWL semantics [Euzenat et al., 2007], [Caracciolo et al., 2008]. Furthermore, nearly all of the approaches seek to learn similarity and equivalence relations between entities across schemas or ontologies; only a few have sought to learn others [Palopoli et al., 1998], [Kotis et al., 2006] and those are restricted to subclass and subsumption. Within recent months, approaches have emerged for subclass relation alignment [Hamdi et al., 2008], [David, 2008] [Jean-Mary and Kabuka, 2008]. Some solutions discover subsumption [Kim et al., 2003], [David, 2008]. Emerging approaches are applying learning techniques to discover disjointness relations [Volker et al., 2007] but much work remains. Therefore, acquiring non equivalence relations between ontological entities remains a research challenge [Zimmerman et al., 2006], [Euzenat and Shvaiko, 2007]. In fact, for the first time in late 2008, the OAEI 2008 report noted that “there are now test cases where non equivalence relations matter” [Caracciolo et al., 2008].
No alignment approach to our knowledge has modeled ontology alignment in a Support Vector Machine. Many believe that alignment techniques will be enhanced with the use of upper ontologies but no system exists yet that uses these [Euzenat and Shvaiko, 2007]. Furthermore, the integration of knowledge across biomedical knowledge sources is lacking due to the complexity of the relationships. There has been some success in applying ontologies for integration in biotechnology [Sahoo et al., 2007], [S. and Bodenreider, 2007], [Elmasri et al., 2007], [Tiffin et al., 2005], [Lambrix and Tan, 2006]; however, the relationships acquired are limited to equivalence and similarity. New methods for clustering gene expression time series data are also emerging with promising results [Das et al., 2007],[Das et al., 2008],[Das et al., 2009]. These approaches are successful in clustering at a coarse level, but further refinement is needed and a semantics-based approach may be complement these approaches.

In 2009, scalability of ontology alignment remains a key challenge and approaches to address the challenge are only starting to emerge in the past two years. In the Ontology Alignment Evaluation Initiative in 2007 and 2008, very few participants aligned large data sets [Euzenat et al., 2007], [Caracciolo et al., 2008]. Scalability continues to be one of the grand challenges of ontology construction and use [Hepp et al., 2008] and scaling ontology alignment in particular remains a research area [Giunchiglia et al., 2009].

Therefore, we see an opportunity to build upon and extend existing techniques. We seek to contribute in three main areas. First, we contribute an approach to extend the information that can be derived in ontology alignment; specifically, we seek to infer non equivalence relations across ontological classes. Second, we contribute
new features and methods to be used in the ontology alignment process. We show that the use of OWL semantics in conjunction with upper ontologies and online linguistic resources can enhance the alignment process and specifically contribute to the discovery of new properties. We model the ontology alignment problem in a Support Vector Machine and show that ontology alignment is enhanced significantly by the use of an SVM. Finally, we investigate approaches to scaling ontology alignment in order to align large-scale biomedical ontologies. We present an alignment algorithm that bounds the processing by selecting optimal subtrees to align.

Before describing our specific contributions, we first present an analysis of ontological entities in the Academic and Biomedical domains in the next Chapter.
Chapter 4

Analysis of Entities in Academic and Biomedical Ontologies

In this chapter, we analyze and characterize the components of the ontologies used in this work. We examine class and relationship names in ontologies in the Academic and Biomedical domains. We discuss other unique characteristics, particularly of the Gene and Mammalian Phenotype ontologies. We offer some examples of how these characteristics can be exploited in ontology alignment. Finally, we also identify other technical considerations that impacted this work.

4.1 Introduction

In this work, we performed ontology alignment in the Academic and Biomedical domains. We randomly selected a set of ontologies by searching ontology repositories on the Web. First, we identified a set of ontologies that describe concepts in the domain of academic events, publications, organizations and people. The ontologies selected were built by independent groups, therefore, we can expect some heterogeneity among the ontologies. Second, we identified a set of biomedical ontologies to use to evaluate our work. We chose to align components of the Gene Ontology (GO) [Ashburner et al., 2000] with the Mammalian Phenotype (MP) ontology [Smith
et al., 2004], part of the Open Biomedical Ontologies (OBO)\textsuperscript{1}. The Gene Ontology provides a controlled vocabulary to describe gene and gene product attributes in any organism [Ashburner et al., 2000]. It consists of three major categories of descriptors: biological processes, molecular functions and cellular components. The Mammalian Phenotype Ontology supports annotation of mammalian phenotypes in the context of mutations, quantitative trait loci and strains that are used as models of human biology and disease [Smith et al., 2004]. The MP ontology is primarily based on experimental studies conducted using mouse as the model organism. We chose to align these two ontologies because: 1) they are well-developed and frequently updated, 2) they are widely used by the biomedical research community, and 3) they have great potential for application in the area of translational research. For example, discoveries in experiments conducted with mice could be linked to various gene products that may share similar GO attributes, which may ultimately contribute to hypothesis generation with regard to human disease development. More information on this idea is presented in Section 8.3.1. Details on the ontologies used in this evaluation are provided in Tables 4.1 and 4.2. Most of the Academic ontologies are fairly small in size and classes and properties consist of commonsense terms. These characteristics made this domain suitable for initial evaluation of our algorithms. We chose to complement our evaluation with a subset of the much larger, more complex GO and MP ontologies. Note that the first column in each table is used to indicate the nickname of each ontology. These nicknames are utilized throughout the rest of the document.

\textsuperscript{1}www.obofoundry.org
Table 4.1: Academic Ontologies Used in Performance Evaluation

<table>
<thead>
<tr>
<th>†</th>
<th>Ontology</th>
<th>Primary Concepts</th>
<th>OWL Classes</th>
<th>OWL Object Properties</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>daml.umbc.edu/ontologies/talk-ont</td>
<td>Event, Place, Person</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>2</td>
<td>ebiquity.umbc.edu/ontology/event.owl</td>
<td>Event, EventSeries</td>
<td>2</td>
<td>5</td>
</tr>
<tr>
<td>3</td>
<td><a href="http://swrc.ontoware.org/ontology/portal">http://swrc.ontoware.org/ontology/portal</a></td>
<td>Conference, Publication</td>
<td>54</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td><a href="http://www.mindswap.org/golbeck/web/www04photo.owl">http://www.mindswap.org/golbeck/web/www04photo.owl</a></td>
<td>Event, Person, Group</td>
<td>19</td>
<td>15</td>
</tr>
<tr>
<td>5</td>
<td><a href="http://ebiquity.umbc.edu/ontology/person.owl">http://ebiquity.umbc.edu/ontology/person.owl</a></td>
<td>Person, Student, Faculty</td>
<td>21</td>
<td>3</td>
</tr>
<tr>
<td>6</td>
<td><a href="http://www.aktors.org/ontology/portal">http://www.aktors.org/ontology/portal</a></td>
<td>Organization, Student</td>
<td>169</td>
<td>171</td>
</tr>
</tbody>
</table>

Table 4.2: Biomedical Ontologies Used in Performance Evaluation

<table>
<thead>
<tr>
<th>†</th>
<th>Ontology</th>
<th>OWL Classes</th>
<th>OWL Object Properties</th>
</tr>
</thead>
<tbody>
<tr>
<td>go</td>
<td>Gene Ontology</td>
<td>27598</td>
<td>4</td>
</tr>
<tr>
<td>mp</td>
<td>Mammalian Phenotype Ontology</td>
<td>6510</td>
<td>1</td>
</tr>
</tbody>
</table>

as other considerations that impacted this work.

4.2 Classes

First, we discuss class characteristics in the Academic domain. Classes in the Academic domain are almost exclusively commonly used nouns. Some examples include Book, Event and Group. In some cases, class names are compound words, with adjectives used to describe the main noun, such as AssociateProfessor and CommitteeMember. In rare cases, a prepositional phrase is used as a class name, as in the cases of InBook and InCollection. In many cases, the ontologies in this domain import external ontologies. When ontologies are imported, namespace tags are used to specify the source of the concept and its semantics. For example, foaf:Person and
Classes in the Biomedical domain consist of, not surprisingly, much less commonly used terms. Some simple examples include *coagulation* and *hemostasis*. Typically, class names involve multiple words, as in the case of *negative regulation of platelet activation*. Some terms are even as complex as *tyrosine biosynthetic process by oxidation of phenylalanine*. Classes in GO tend to be descriptive in nature, since the purpose of GO is to provide a controlled set of terminology to describe gene and gene product attributes in organisms. On the other hand, the purpose of the MP ontology is to provide a standard for annotating mammalian phenotypes. The contributors to MP tend to be researchers who are observing abnormalities in organisms. Therefore, the most specific classes in MP tend to reflect abnormalities. For example, *abnormal platelet activation, excessive ear growth* and *delayed bone ossification* are example class names in MP. Mapping GO and MP offered many examples of subclass relationships, such as *abnormal platelet activation* in MP mapped as a subclass of *platelet activation* in GO.

Given these characteristics of class names in the biomedical domain, bag-of-word type features were extracted for use in learning by the SVM. This was done to maximize performance in ontology alignment. For example, the number of synonyms in class names cross-ontology is counted as a feature to be considered in alignment. So

---

*foaf:* is a namespace tag that refers to the Friend of a Friend ontology (http://www.foaf-project.org/). This is an ontology designed to describe people, their activities and relations to one another.
processing of two classes, such as abnormal platelet activation and platelet activation would yield a feature of 2 synonyms shared between class names. This approach was effective in detecting relationships cross-ontology in this particular set of ontologies. (More details are provided in Chapter 6.)

We also exploited the fact that MP tends to describe abnormal phenotypes. Resultant linguistic features were extracted. For example, if two class names match, except that one is preceded by an adjective, such as the case of platelet activation and abnormal platelet activation, then a feature is extracted that suggests that the latter is likely a hyponym of the former.

In addition to biological class names, GO and MP include the following special classes.

- oboInOwl:DbXref
- oboInOwl:Definition
- oboInOwl:Subset
- oboInOwl:Synonym
- oboInOwl:SynonymType
- oboInOwl:ObsoleteClass

oboInOwl:DbXref is used to cross reference to external databases. For example, a term in GO can be cross referenced to an external database. oboInOwl:Definition is used to define the meaning of a class in natural language. For example, the class platelet activation in GO has the following definition:
“A series of progressive, overlapping events triggered by exposure of the platelets to subendothelial tissue. These events include shape change, adhesiveness, aggregation, and release reactions. When carried through to completion, these events lead to the formation of a stable hemostatic plug.”

This is another potential characteristic of biomedical ontologies that could be exploited for alignment. Natural language processing techniques could be used to derive more information from the description.

The relationship oboInOwl:Subset indicates membership in a subclass. That is, if X oboInOwl:hasSubset oboInOwl:Subset Y, then Y is a member of X. This is another method to include subordinate class relationships in OBO ontologies and in GO.

oboInOwl:Synonym is a class that allows specification of a synonym of another class. However, in use, GO synonyms are not always synonyms in the strictest sense of the word. Instead, a GO synonym may be broader or narrower than the term string; it may be a related phrase; it may be alternative wording, spelling or use a different system of nomenclature; or it may be a true synonym. The class oboInOwl:SynonymType allows specification of the type of synonym. Choices for SynonymType include exact, broader, narrower, and related. If oboInOwl:Synonym is used in the alignment process, the type of synonym would need to be considered.

oboInOwl:ObsoleteClass is a class used to denote classes that are no longer in use.

3Note that more information on special relationships, such as oboInOwl:hasSubset are described in more detail later in this section.
Table 4.3: Relationship pattern frequency in Academic ontologies used in this work

<table>
<thead>
<tr>
<th>Pattern</th>
<th>Examples</th>
<th>Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>verb phrase</td>
<td>hasLocation, includedInPublication, isCapitalOf</td>
<td>49%</td>
</tr>
<tr>
<td>noun</td>
<td>location, eventProduct</td>
<td>27%</td>
</tr>
<tr>
<td>noun phrase</td>
<td>atEvent, eventAttended, governmentOfCountry, headOf</td>
<td>19%</td>
</tr>
<tr>
<td>verb</td>
<td>develops, employs</td>
<td>4%</td>
</tr>
<tr>
<td>sentence</td>
<td>projectInvolvesOrganizationalUnit</td>
<td>1%</td>
</tr>
</tbody>
</table>

4.3 Relationships

Relationship names in ontologies found on the Web are unconstrained and can essentially be anything, as is the case with class names. The definition of these is the responsibility of the author of the ontology. In the Academic ontologies used in this work, there was great variety in the definition and usage of relationships. Some example relationships from the Academic ontologies include developedBy, develops, author, edited-by, location, and hasLocation. As can be seen in these examples, relationships often contain more than one word. Relationships that are composed of multiple words are separated in a number of ways. Words are separated using dashes and underlines, such as cited-by and cited_by. They are also separated using capital letters, as in citedBy. Though the relationships are very diverse, they can be classified into a number of distinct patterns. These patterns are shown in Table 4.3.

Verb phrase is the most common pattern found in the Academic ontologies used in this work. The most prevalent form of this pattern is the relationship hasX, where X is a noun. Some examples include hasParticipant and hasMember. This
relationship can be exploited to determine ontology alignments. For example, any relationship of the form hasX, where X is a subclass of a Person, can be used to align with synonyms or hyponyms of Person in a second ontology. For example, the property hasParticipant in one ontology can be aligned to Person and its subclasses in a second ontology.

Noun phrase is also a common pattern found. Some examples include location and eventProduct. The semantics of these types of relationships is typically that of the common verb phrase hasX described above. For example, by using the relationship location, the ontology author typically means hasLocation. So this type of pattern can be exploited in the same way as described above. More information on how patterns within ontological relationships were exploited is described in Chapters 5 and 6.

Next, we discuss specific properties in the Gene Ontology (GO). Concepts in GO can be linked with five relationships: is_a, part_of, regulates, positively_regulates and negatively_regulates. The is_a relationship is a subclass relationship, defined as follows. If X is_a Y then X is a subclass of Y. Subclass here means a specialization of the more general class and is transitive in nature. The part_of relationship is defined in GO as follows. If X part_of Y, then when X is present it is always part of Y but X need not be present. An example of this relationship from [Ashburner et al., 2000] is shown in Figure 4.1. In this case, when a periplasmic flagellum (GO_0055040) is present, it is always part_of a periplasmic space (GO_0042597). However, every

---

4Note that hasParticipant is not always an accurate alignment to subclasses of Person in the purest semantic interpretation of hasParticipant. However, we argue that for the purposes of alignment of ontologies, particularly in biomedicine, it may still be useful to acquire such alignments.
Figure 4.1: Example of part_of relationship in GO, expressed in OWL.

periplasmic space does not necessarily have a periplasmic flagellum [Ashburner et al., 2000]. This example is expressed in OWL as a subclass restriction.

The regulates, positively_regulates and negatively_regulates relationships describe interactions between biological processes and other biological processes, molecular functions or biological qualities. When a biological process X regulates a function or a process Y, it modulates the occurrence of Y. For example, when regulation of transcription occurs, it always alters the rate, extent or frequency at which a gene is transcribed.

Both is_a and part_of are transitive relationships in GO. The regulates relationships are transitive over both the part_of and is_a relationships. This means that if X is_a Y or if X is part_of Y, then any process that regulates Y also regulates X. These relations can be exploited in the alignment process.

As described earlier, the Mammalian Phenotype (MP) ontology is part of the
Open Biomedical Ontologies (OBO). OBO defines a large set of relationships, as shown in Table 4.4. Of these, only is_a and part_of are used in MP. The is_a relationship is the most used relationship; part_of does not appear that often in MP. However, we should consider all possible OBO relationships since we seek to develop a generic solution.

The OBO ontologies also contain a set of special annotation properties, as follows.

- oboInOwl:hasAlternativeId
- oboInOwl:hasDate
- oboInOwl:hasVersion
- oboInOwl:hasDbXref
- oboInOwl:hasDefaultNamespace
- oboInOwl:hasOBONamespace
- oboInOwl:hasDefinition
- oboInOwl:hasSynonym
- oboInOwl:hasExactSynonym
- oboInOwl:hasNarrowSynonym
- oboInOwl:hasBroadSynonym
- oboInOwl:hasRelatedSynonym
<table>
<thead>
<tr>
<th>Relationship</th>
<th>Transitive</th>
<th>Symmetric</th>
<th>Reflexive</th>
<th>Anti-Symmetric</th>
</tr>
</thead>
<tbody>
<tr>
<td>is_a</td>
<td>x</td>
<td>x</td>
<td></td>
<td>x</td>
</tr>
<tr>
<td>part_of</td>
<td>x</td>
<td>x</td>
<td></td>
<td>x</td>
</tr>
<tr>
<td>integral_part_of</td>
<td>x</td>
<td>x</td>
<td></td>
<td>x</td>
</tr>
<tr>
<td>proper_part_of</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>located_in</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>contained_in</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>adjacent_to</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>transformation_of</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>derives_from</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>predeced_by</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>has_participant</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>has_agent</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>instance_of</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- `oboInOwl:hasSynonymType`
- `oboInOwl:hasSubset`
- `oboInOwl:hasURI`
- `oboInOwl:isCyclic`
- `oboInOwl:inSubset`
- `oboInOwl:savedBy`
- `oboInOwl:replacedBy`
- `oboInOwl:consider`
Some of these properties are strictly for annotation and do not offer much in the way of information for ontology alignment, such as hasDate, hasVersion, hasDefaultNamespace and hasURI. Others however, offer some additional useful information such as hasDefinition and hasSynonym. These properties are used in conjunction with the special classes oboInOwl:Definition and oboInOwl:Synonym, described earlier in this section. The properties hasExactSynonym, hasNarrowSynonym, hasBroadSynonym, hasRelatedSynonym are essentially sub properties\(^5\) of hasSynonym. These properties can be used to define the synonymy relation between classes more specifically. This type of information could be very useful in ontology alignment.

Other properties are designed for use by the subject matter expert. For example, oboInOwl:consider allows specification of a term which may be an appropriate substitute for an obsolete term, but needs to be looked at carefully by a human expert before the replacement is done. Another example is oboInOwl:replacedBy. This property specifies a term that replaces an obsolete term.

### 4.4 Other Considerations

It should be noted that GO and the OBO ontologies represent class names as identifiers preceeded with the ontology name. For example, as seen in Figure 4.1, the class GO:0055040 is the class name that represents the concept *periplasmic flagellum*. The more meaningful name of the class is represented in the rdfs:label property in the

\(^5\)It should be noted that to remain in OWL DL, sub properties are not formally declared in GO, though that is certainly the intent [http://www.geneontology.org/formats/oboInOwl:hasNarrowSynonym].
ontology. So the class GO:0055040 has the label *periplasmic flagellum*. MP and other OBO ontologies are designed the same way. Each class name is preceded by MP_ or appropriate prefix. Therefore additional processing is necessary in the biomedical domain to extract the meaningful label of each class.

The ontologies used in the Academic domain use meaningful class names and often do not contain labels. When labels are present, they are used in a variety of ways. For example, a label for the class *AssociateProfessor* is as *Associate Professor*; so the label distinguishes the words of the class name in this example.

Many ontologies, including those we used from the Academic domain, use prefixes to denote namespaces. This is particularly the case when ontologies are imported. Therefore, it is typical to see class names such as *p1:Event* and *foaf:Person*. In this work, all class and property names were normalized prior to processing.

In the next Chapter, we present patterns for alignment using non equivalence relations.
Chapter 5

Patterns for Alignment using Non Equivalence Relations

In this chapter, we present a set of patterns designed to acquire non equivalence relationships between ontological classes. The patterns utilize the semantics of OWL and external knowledge sources, including an upper ontology, to align ontologies using the relationships defined in Chapter 2. These relationships include subclass and superclass as well as inferred relations cross-ontology. The external knowledge sources are domain-independent so that use of this approach is not restricted to any particular domain.

5.1 Evidence Primitives

Our initial strategy for acquiring non equivalence relationships across ontological entities was to apply reasoning patterns that use the semantics of OWL in conjunction with knowledge in the upper ontology OpenCyc and the popular linguistic resource WordNet. OpenCyc is the open source version of the Cyc upper ontology, one of the most comprehensive general knowledge bases and commonsense reasoning engines, developed by Cycorp. Cyc contains hundreds of thousands of terms and millions of assertions relating the terms, forming an upper ontology intended to represent commonsense human knowledge. OpenCyc represents a subset of the terms
in Cyc, and provides a foundational set of basic commonsense knowledge. WordNet is a large database of English, in which nouns, verbs, adjectives and adverbs are grouped into sets of cognitive synonyms (called synsets), each expressing a distinct concept. Synsets are interlinked by taxonomic relations, including hyponymy, hypernymy, meronymy, holonymy and more. WordNet 3.0 contains nearly 207,000 synset pairs.

In this work, OpenCyc and WordNet are consulted to identify direct relationships between class pairs, such as broaderThan and narrowerThan in the former and synonymy, hyponymy and hypernymy in the latter. The sources are used to identify direct relationships between classes in a pair but are also used in the patterns described later in this Chapter and in Chapter 6. We used other relations from OpenCyc, including alias, for example. To access the information we use the OpenCyc 1.0.2 API. We used the following relationships from WordNet: seeAlso, hyponym, hypernym, similarity. To access the knowledge in WordNet, we use the Java WordNet Library (JWNL).

Each unit of knowledge obtained from OpenCyc and WordNet as well as the semantics of the OWL representation is viewed as an “evidence primitive”. Each evidence primitive is a relationship that can occur within an ontology or across ontologies. Combining these evidence primitives in different ways allows discovery of relationships across ontological entities. We group the evidence primitives into sets, as shown in Table 5.1. For example, knowledge that suggests similarity or equivalence

---

1 OpenCyc, http://www.opencyc.org/
of concepts within a single ontology is categorized into set $s_0$. Grouping of evidence allows us to simplify the patterns, which are described further in the next Section. The set of evidence primitives and the source for each used in this work is provided in Table 5.2.

### 5.2 Patterns of Evidence for Alignment

To acquire relations between ontological entities, we define a set of patterns of evidence that use the semantics of OWL with knowledge from OpenCyc and WordNet. Patterns were defined to acquire subclass, superclass and inferred relations cross-ontology using relations specified within the original ontologies. Three patterns were defined for each relationship as follows. We name each pattern based on the key cross-ontological relationship upon which it depends.

- Subclass Pattern 1, based on synonymy

<table>
<thead>
<tr>
<th>Set Name</th>
<th>Evidence Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$s_0$</td>
<td>Symmetric relation that suggests similarity or equivalence within an ontology</td>
</tr>
<tr>
<td>$s_1$</td>
<td>Symmetric relation that suggests similarity or equivalence across ontologies</td>
</tr>
<tr>
<td>$h_0$</td>
<td>Directed relation that specifies subclass relation within an ontology</td>
</tr>
<tr>
<td>$h_1$</td>
<td>Directed relation that specifies subclass relation across ontologies</td>
</tr>
<tr>
<td>$h_2$</td>
<td>Directed relation that specifies superclass relation within an ontology</td>
</tr>
<tr>
<td>$h_3$</td>
<td>Directed relation that specifies superclass relation across ontologies</td>
</tr>
</tbody>
</table>

Table 5.1: Evidence primitive types
Table 5.2: Evidence primitives, sources and set membership

<table>
<thead>
<tr>
<th>Evidence Primitive</th>
<th>Source</th>
<th>Defined in set</th>
</tr>
</thead>
<tbody>
<tr>
<td>subclass</td>
<td>OWL</td>
<td>$h_0$</td>
</tr>
<tr>
<td>equivalentClass</td>
<td>OWL</td>
<td>$s_0$</td>
</tr>
<tr>
<td>“see also”</td>
<td>WordNet</td>
<td>$s_1$</td>
</tr>
<tr>
<td>hyponymy</td>
<td>WordNet</td>
<td>$h_2$</td>
</tr>
<tr>
<td>hypernymy</td>
<td>WordNet</td>
<td>$h_1$</td>
</tr>
<tr>
<td>similarity</td>
<td>WordNet</td>
<td>$s_1$</td>
</tr>
<tr>
<td>alias</td>
<td>OpenCyc</td>
<td>$s_1$</td>
</tr>
<tr>
<td>moreGeneral</td>
<td>OpenCyc</td>
<td>$h_2$</td>
</tr>
<tr>
<td>moreSpecific</td>
<td>OpenCyc</td>
<td>$h_1$</td>
</tr>
</tbody>
</table>

- Subclass Pattern 2, based on hyponymy
- Subclass Pattern 3, based on hypernymy
- Superclass Pattern 1, based on synonymy
- Superclass Pattern 2, based on hyponymy
- Superclass Pattern 3, based on hypernymy
- Inferred Relation Pattern 1, based on synonymy
- Inferred Relation Pattern 2, based on hyponymy
- Inferred Relation Pattern 3, based on hypernymy

First, we discuss the patterns for subclass. Subclass Pattern 1 is based on the cross ontological relation of synonymy and models the following case. Within ontology$_1$, there are 2 classes $c_1$ and $c_2$ that share a relation in $h_0$, and, in ontology$_2$,
there is a class $c_3$ that shares a relation in the set $s_1$ with $c_1$. We hypothesize that this evidence suggests that $c_2$ is a subclass of $c_3$. This pattern is shown in Figure 5.1.

An example of Subclass Pattern 1 is shown in Figure 5.2. In this case, within ontology$_1$, there is a class *PeriodontalBoneResportion* specified to be a subclass of *BoneResportion*. Using a variety of techniques, such as string distance and consultation of external sources (these techniques are discussed in the next Chapter), we discover that *BoneResportion* is synonymous with *BoneLoss*, a class in ontology$_2$. We submit that this provides evidence that *PeriodontalBoneResportion* is likely a subclass of *BoneLoss*.

Subclass Pattern 2 models a similar case except that this pattern is based on a hyponymic relation cross-ontology. Within ontology$_1$, there are 2 classes $c_1$ and $c_2$ that share a relation in $h_0$; in ontology$_2$, there is a class $c_3$ that shares a relation in the set $h_1$ with $c_1$. This evidence suggests that $c_2$ is a subclass of $c_3$. This pattern is shown in Figure 5.3.

Figure 5.1: Subclass Pattern 1, based on Synonymy.
Subclass Pattern 3 is very similar to Pattern 2, except that the pattern uses superclass relationships, that is, relations in \( h_2 \) cross ontology. In this case, the same relationships exist between \( c_1 \) and \( c_2 \) in ontology\(_1\). In ontology\(_2\), there is a class \( c_3 \) that is a superclass of \( c_1 \); that is, \( c_3 \) shares a relation in the set \( h_2 \) with \( c_1 \). We propose that this evidence suggests that \( c_2 \) is a subclass of \( c_3 \). This pattern is shown in Figure 5.4. Note that we distinguish between the directed relationships \((c_1, h_1, c_3)\) and \((c_3, h_2, c_1)\) because subclass and superclass relationships cross-ontology are discovered.
as separate evidence primitives and may be very different; in fact, we found that to be the case. That is, it is possible that subclass and superclass relationships are explicitly defined and thus we must be able to detect both.

Three patterns to detect superclass relationships were also developed. These followed the same approach as subclass patterns, in which the patterns rely on synonymy, hyponymy and hypernymy. Since the superclass patterns follow the same approach as subclass patterns, these are not discussed further. Further details on these patterns can be found in [Stoutenburg and Kalita, 2008], [Stoutenburg et al., 2009].

The patterns to detect Inferred relations also use a variety of evidence primitives to acquire relations cross-ontology that are specified in the original ontologies. First, Inferred Relation Pattern 1 detects the following case. Within ontology$_1$, there are 2 classes $c_1$ and $c_2$ that share some relation $p$ defined in the original ontology; and, in ontology$_2$, there is a class $c_3$ that shares a relation in the set $s_1$ with $c_2$. This evidence
Figure 5.5: Pattern 1 for Inferred Relations, based on synonymy cross-ontology.

suggests that \( c_1 \) and \( c_3 \) share that same property \( p \) cross-ontology. This pattern is shown in Figure 5.5.

Recall that an example of this pattern was presented in Figure 1.1 of Chapter 1. Recall that since it is given that \((\text{PlateletActivation}, \text{partOf}, \text{BloodCoagulation})\) in ontology\(_1\), and \((\text{BloodCoagulation}, \text{synonymy}, \text{Clotting})\), it is inferred that \((\text{PlateletActivation}, \text{partOf}, \text{Clotting})\) is one likely alignment between ontology\(_1\) and ontology\(_2\).

Inferred Relation Patterns 2 and 3 are similar, except that we employ the relations in \( h_1 \) and \( h_2 \), respectively, to recommend the relation \( p \) between classes \( c_1 \) and \( c_3 \). Inferred Relation Pattern 2 is shown in Figure 5.6.
5.3 Evaluation

For initial testing, we randomly selected a set of ontologies by searching ontology repositories on the Web. As described in Chapter 4, we chose ontologies that describe concepts in the domains of academic events, publications, organizations and people for initial evaluation of patterns for alignment. We purposely chose two ontologies that were small to allow for easy preliminary testing, then complemented those with larger ontologies in the related domain. The ontologies used are shown in Table 4.1 in Chapter 4.

We developed a reference alignment (as defined in Euzenat [2007]) of mapping results semi-automatically; the reference alignments were manually specified, but the process of constructing data sets was aided with software for efficiency. Precision
and recall were then automatically measured against the reference alignment. We restricted expected ontological mappings to the semantics expressed in each ontology; that is, we did not expect to acquire more information than is expressed in each ontology. For example, if the class Student is not declared to be a subclass of Person, then we could not expect our algorithms to map Student with properties associated with Person, as would occur in Subclass Pattern 1 (though that relation would be expected to be discovered in other patterns, for example, Inferred Relation r Pattern 2). We selected pairs of ontologies to map; the mapping is obviously directional, so different evaluations were done with ontology pairs 1,4 and 4,1, for example. Pairs were selected to ensure that as many unique combinations of mappings as possible were tested. Some pairs were not tested as they either yielded an empty reference alignment or yielded a significant number of similar mappings as in other pairs; this depended on the characteristics of the ontologies and the pattern applied.

We used precision, recall and f-score to measure the exactness, completeness and overall performance of the alignment algorithms. Precision is calculated as follows:

$$P = \frac{tp}{tp + fp}$$

where:

- $tp =$ the number of true positives, that is, the number of positive examples in the reference alignment that were classified as positive by the algorithm,

- $fp =$ the number of false positives, that is, the number of positive examples in the reference alignment that were classified as negative by the algorithm,

- $tn =$ the number of true negatives, that is, the number of negative examples in
the reference alignment that were classified as negative by the algorithm, and

- $fn =$ the number of false negatives, or the number of negative examples in the
  reference alignment that were classified as positive by the algorithm.

Recall is calculated as:

$$R = \frac{tp}{tp + fn}$$

We also present micro and macro averages of precision and recall. The micro
average is calculated using the precision and recall formulas above over the total set
of $tp, fp, tn$ and $fn$. The macro average is calculated by averaging the precision and
recall calculations of each ontology pair.

In some areas of the evaluation, we computed f-score. The f-score is calculated
as follows.

$$F = \frac{2PR}{P + R}$$

To visually see the results, we use two techniques; first, we use graphs that
plot the sensitivity of the algorithm (true positive rate) vs. the specificity of the
algorithm (false positive rate.) This technique is used in classic Receiver Operator
Curves (ROC). In our case, the alignment prediction is discrete, so instead of a curve,
the data appears as a scatter plot. A legend of the scatter plots is shown in Figure 5.7.
Performance results that appear in the upper left hand corner of the graph are perfect;
therefore the further above the line representing $x = y$, the better the performance.

The second technique used is a modified Detection Error Tradeoff (DET) curve
[Martin et al., 1997]. DET curves plot error rates on the graph, typically false alarm
probability vs. miss probability. In DET curves, points near the bottom left part of
the graph represent the lowest error and therefore the best performance. By using the DET curve, the errors can be seen more clearly and there is less gathering of data points as in the ROC curves [Martin et al., 1997]. Instead of plotting false alarm probability vs. miss probability, we plot the false positive rate vs. the false negative rate in our evaluation.

5.4 Experimental Results

As we evaluated the performance of the patterns, we found that performance can vary widely depending on the external knowledge sources utilized. Therefore we performed multiple tests of each ontology pair, utilizing WordNet as the sole external knowledge source, OpenCyc as the sole knowledge source and finally, utilizing both knowledge sources. We report the precision, recall and f-score of each test.
5.4.1 Patterns to Align using Subclass Relations

In this section, we present the evaluation results of the three patterns to align using subclass relations. The accuracy of Subclass Pattern 1 is promising, as can be seen in Table 5.3. The pattern is performing well in most cases; in the cases of lower precision, such as pairs 4,3, 4,5 and 4,6, analysis showed that this is a result of unanticipated relationships returned from WordNet. For example, in the case of ontology pair 4,5, Presenter and Sponsor are identified in WordNet as synonyms. Unanticipated relationships lead to lower precision in the patterns. This can be seen in pairs 4,3 and 4,5 where OpenCyc performed much better. Also note that for pair 4,6, recall is fairly low. We found that the class pairs missed included terms such as Giving-A-Talk and Social-Gathering. This suggests that recall could be improved by performing additional pre-processing of class name strings. This is discussed further later in this Chapter.

In the case of Subclass Pattern 1, there is not a significant difference in the performance depending on the knowledge source used, on average (though for pairs 4,3 and 4,5, OpenCyc performs better). Recall that Subclass Pattern 1 depends on synonymy cross-ontology; therefore, these results suggest that WordNet and OpenCyc perform fairly well in providing synonymy relationships. The performance of Subclass Pattern 1 is shown visually in Figure 5.8. This visually shows that Subclass Pattern 1 is performing quite well, with fairly high true positive and low false positive rates. A DET curve for Subclass Pattern 1 is provided in Figure 5.9. This shows that the performance is fairly close to the 0.0 axis, with only a few outliers, an indicator of
Table 5.3: Subclass Pattern 1 Evaluation Results

<table>
<thead>
<tr>
<th>Ontology Pair</th>
<th>WordNet</th>
<th>OpenCyc</th>
<th>WordNet and OpenCyc</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Prec</td>
<td>Rec</td>
<td>F</td>
</tr>
<tr>
<td>4,1</td>
<td>.83</td>
<td>1.00</td>
<td>.91</td>
</tr>
<tr>
<td>4,2</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
</tr>
<tr>
<td>3,4</td>
<td>.98</td>
<td>.87</td>
<td>.92</td>
</tr>
<tr>
<td>4,3</td>
<td>.71</td>
<td>.77</td>
<td>.74</td>
</tr>
<tr>
<td>4,5</td>
<td>.63</td>
<td>1.00</td>
<td>.77</td>
</tr>
<tr>
<td>5,4</td>
<td>.95</td>
<td>.98</td>
<td>.96</td>
</tr>
<tr>
<td>4,6</td>
<td>.71</td>
<td>.63</td>
<td>.67</td>
</tr>
<tr>
<td>6,4</td>
<td>.92</td>
<td>.82</td>
<td>.87</td>
</tr>
<tr>
<td>Micro Avg</td>
<td>.90</td>
<td>.85</td>
<td>.87</td>
</tr>
<tr>
<td>Macro Avg</td>
<td>.87</td>
<td>.91</td>
<td>.89</td>
</tr>
</tbody>
</table>

fairly low error rates.

The evaluation of Subclass Pattern 2 is shown in Table 5.4. This pattern is also fairly promising, though it is less accurate overall than Subclass Pattern 1. For example, in ontology pair 4,5 precision is quite low. Analysis shows that this was because WordNet returns the hyponym pairs that were not expected. For example, WordNet returned Person, Collaborator and Presenter, Sponsor as hyponym pairs. Since there are a significant number of subclasses of Person in ontology 4, for example, this unexpected hyponym relation had broader consequences in the overall accuracy of the results. This suggests that we should modify the patterns to consider additional knowledge sources and weigh each evidence primitive, or feature, according to a set of known contributing features. (In the next Chapter, we describe how a machine
The ROC scatterplot is shown in Figure 5.10. This shows that the outliers, those alignments with the highest false positive rates, occur when WordNet is used as the external knowledge source. When WordNet is used with OpenCyc, WordNet degrades the performance. The error rates of Subclass Pattern 2 are shown in Figure 5.11. This diagram shows that Subclass Pattern 2 results in a very low rate of false alarms overall with only a few outliers. The outliers are primarily due to WordNet, as can be seen in the diagram. False negative rates are a bit higher, indicating a high miss rate, that is, a lower precision.

Despite these challenges, overall performance is somewhat encouraging, partic-
Figure 5.9: DET scatterplot showing error rates of Subclass Pattern 1.

ularly considering the simplicity of the patterns and ease of implementation. This is particularly true of results of aligning pair 6,4; in this case, the patterns delivered fairly strong performance on one of the larger sets of ontology pairs in this domain.

One of the most striking findings in the analysis of Subclass Pattern 2 is the significant difference in results when using OpenCyc vs. WordNet as external knowledge sources. Use of OpenCyc as the sole knowledge source results in superior accuracy in every alignment pair. Recall that Subclass Pattern 2 relies on hyponymic and subclass relationships cross-ontology. The evaluation of Subclass Pattern 2 suggests that OpenCyc is by far, a superior source of hyponymic (specifically, narrowerThan) relationships. The precision of alignment when using WordNet is significantly lower, and therefore it appears that WordNet is not a strong source of hyponymic relationships. Note that use of both knowledge sources degrades performance. In addition, it is important to note that OpenCyc has delivered valuable narrowerThan relationships
in this testing, but this testing was performed in a simplistic domain in which class names are commonsense terms. Therefore, these patterns, as written, would not be expected to perform well in more complex domains with class names that are not commonsense terms.

Subclass Pattern 3 is the poorest performer of the three subclass patterns when WordNet is used as the external knowledge source. The evaluation of Subclass Pattern 3 is shown in Table 5.5. Again, as in the case of Pattern 2, we see that WordNet is not a strong knowledge source for hypernymic relations. In this case, WordNet is returning hypernym pairs that are unexpected. For example, WordNet returned \{Workshop, Place\} as hypernym pairs. Since there are a significant number of subclasses of Place in other ontologies, this unexpected hypernym relation had broader consequences
Figure 5.10: ROC scatterplot showing performance of Subclass Pattern 2 which relies on hyponymy cross-ontology. In the overall accuracy of the results. There were numerous other examples very similar to this. However, OpenCyc delivers outstanding performance, suggesting that OpenCyc is a very good source for hypernymic relationships, in addition to hyponymic relationships as was observed in the evaluation of Subclass Pattern 2. The ROC scatter diagram for Subclass Pattern 3 is provided in Figure 5.12. This diagram highlights the fact that WordNet is the source of significant false positives. Overall, with the use of OpenCyc, this pattern performs very well, with near zero false hits, and precision in all cases. These observations are confirmed in the DET scatter diagram in Figure 5.13. This diagram shows clearly that WordNet is the source of error and OpenCyc performs with near zero false positive rate.
Figure 5.11: DET scatterplot showing error rates of Subclass Pattern 2.

The overall performance of the patterns to detect subclass relationships between ontologies is shown in Figure 5.14. This diagram shows that of the three patterns, pattern 3 performs the worst, when WordNet is used as the sole knowledge source of hypernymic relations. Overall, however, the patterns perform quite well, with low false positive rates and fairly high true positive rates in most cases. It is important to note that the strong performance of OpenCyc as an external knowledge source in Subclass Patterns 2 and 3 is limited by the fact that the class names used in this domain are commonsense terms. Therefore, we can expect that in domains with more complex class names, such as the biomedical domain, OpenCyc as is will not be useful as used in these patterns. This suggests that additional features will be required to perform accurate alignments in the biomedical domain.
5.4.2 Patterns to Align using Superclass Relations

The evaluation results of the patterns to align using superclass relations is shown in Figure 5.15. In this figure, we chart the f-score of the alignment results for each ontology pair. We compare the performance of alignment when using WordNet, OpenCyc and both knowledge sources. Performance of alignment with superclass pattern 1 is very encouraging. Note that ontology pair 6,4 resulted in the lowest f-score. This was a result of very low recall; pair 6,4 was aligned with 100% precision but only 36% recall. An analysis revealed that the pattern was simply too restrictive to identify additional alignments. This suggests that additional features should be added to complement the patterns. Note that in superclass pattern 1, both WordNet and OpenCyc perform similarly and the pattern delivers an average precision of well over .90 and recall of over .80.

<table>
<thead>
<tr>
<th>Ontology Pair</th>
<th>WordNet Prec</th>
<th>Rec</th>
<th>F</th>
<th>OpenCyc Prec</th>
<th>Rec</th>
<th>F</th>
<th>WordNet and OpenCyc Prec</th>
<th>Rec</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>4,1</td>
<td>.59</td>
<td>1.00</td>
<td>.74</td>
<td>1.00</td>
<td>.92</td>
<td>.96</td>
<td>.59</td>
<td>1.00</td>
<td>.74</td>
</tr>
<tr>
<td>4,2</td>
<td>.56</td>
<td>.83</td>
<td>.67</td>
<td>1.00</td>
<td>.83</td>
<td>.91</td>
<td>.56</td>
<td>.83</td>
<td>.67</td>
</tr>
<tr>
<td>3,4</td>
<td>.32</td>
<td>.90</td>
<td>.47</td>
<td>1.00</td>
<td>.70</td>
<td>.82</td>
<td>.32</td>
<td>.76</td>
<td>.45</td>
</tr>
<tr>
<td>4,3</td>
<td>.33</td>
<td>.94</td>
<td>.49</td>
<td>1.00</td>
<td>.75</td>
<td>.86</td>
<td>.33</td>
<td>.94</td>
<td>.49</td>
</tr>
<tr>
<td>4,5</td>
<td>.50</td>
<td>1.00</td>
<td>.67</td>
<td>.83</td>
<td>1.00</td>
<td>.91</td>
<td>.50</td>
<td>1.00</td>
<td>.67</td>
</tr>
<tr>
<td>5,4</td>
<td>.33</td>
<td>1.00</td>
<td>.49</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>.33</td>
<td>1.00</td>
<td>.49</td>
</tr>
<tr>
<td>4,6</td>
<td>.22</td>
<td>.63</td>
<td>.33</td>
<td>.96</td>
<td>.83</td>
<td>.89</td>
<td>.27</td>
<td>.83</td>
<td>.41</td>
</tr>
<tr>
<td>Micro Avg</td>
<td>.16</td>
<td>.91</td>
<td>.27</td>
<td>.98</td>
<td>.81</td>
<td>.89</td>
<td>.34</td>
<td>.86</td>
<td>.49</td>
</tr>
<tr>
<td>Macro Avg</td>
<td>.37</td>
<td>.91</td>
<td>.52</td>
<td>.97</td>
<td>.86</td>
<td>.91</td>
<td>.41</td>
<td>.91</td>
<td>.57</td>
</tr>
</tbody>
</table>
Figure 5.12: ROC scatterplot showing performance of Subclass Pattern 3 which relies on hypernymy cross-ontology.

In the performance of pattern 2 in Figure 5.15, the difference in performance of OpenCyc and WordNet is striking. OpenCyc performs much better. Recall that pattern 2 relies on hyponymy relations cross-ontology. As discussed earlier, OpenCyc performs better than WordNet as a provider of hyponymic relations. This difference in performance is especially notable in pattern 2. When both knowledge sources are used together, performance degrades.

Pattern 3 for aligning with superclass relations performed well. In this case, OpenCyc still performs better than WordNet, particularly in the alignment of pairs 4,1, 4,2 and 5,4. But there is not as striking a difference as in pattern 2. Overall, subclass 3 performs on average with 96% precision and 88% recall when OpenCyc is
used as the sole knowledge source. With WordNet as one of or the sole knowledge source, the pattern performs at just 67% precision and 82% recall. Detailed evaluation results of the performance of superclass patterns are shown in Table 5.6.
Figure 5.14: Overall performance of patterns to align using subclass relations.
Figure 5.15: Alignment accuracy of superclass patterns.
Table 5.6: Superclass Pattern Evaluation Results

<table>
<thead>
<tr>
<th>Ontology Pair</th>
<th>Superclass Pattern 1</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>WordNet</td>
<td>OpenCyc</td>
<td>Both</td>
<td>Prec</td>
<td>Rec</td>
<td>Prec</td>
<td>Rec</td>
<td>Prec</td>
<td>Rec</td>
<td>Prec</td>
<td>Rec</td>
<td>Prec</td>
<td>Rec</td>
<td>Prec</td>
<td>Rec</td>
</tr>
<tr>
<td>1,4</td>
<td>.92</td>
<td>.85</td>
<td>.92</td>
<td>.85</td>
<td>.92</td>
<td>.85</td>
<td>.55</td>
<td>.92</td>
<td>.85</td>
<td>.55</td>
<td>.92</td>
<td>.69</td>
<td>.85</td>
<td>.92</td>
<td>.85</td>
</tr>
<tr>
<td>4,1</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>.50</td>
<td>1.00</td>
<td>1.00</td>
<td>.50</td>
<td>1.00</td>
<td>.50</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
</tr>
<tr>
<td>2,4</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>.56</td>
<td>1.00</td>
<td>1.00</td>
<td>.56</td>
<td>1.00</td>
<td>.56</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
</tr>
<tr>
<td>3,4</td>
<td>1.00</td>
<td>.75</td>
<td>1.00</td>
<td>.75</td>
<td>1.00</td>
<td>.75</td>
<td>.29</td>
<td>.81</td>
<td>1.00</td>
<td>.29</td>
<td>.81</td>
<td>.62</td>
<td>.81</td>
<td>1.00</td>
<td>.75</td>
</tr>
<tr>
<td>4,3</td>
<td>.92</td>
<td>.80</td>
<td>.92</td>
<td>.80</td>
<td>.92</td>
<td>.80</td>
<td>.20</td>
<td>.80</td>
<td>.71</td>
<td>.19</td>
<td>.83</td>
<td>.73</td>
<td>.80</td>
<td>.71</td>
<td>.83</td>
</tr>
<tr>
<td>4,5</td>
<td>1.00</td>
<td>.95</td>
<td>1.00</td>
<td>.95</td>
<td>1.00</td>
<td>.95</td>
<td>.33</td>
<td>.95</td>
<td>1.00</td>
<td>.33</td>
<td>.95</td>
<td>1.00</td>
<td>.95</td>
<td>1.00</td>
<td>.95</td>
</tr>
<tr>
<td>5,4</td>
<td>.86</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>.86</td>
<td>1.00</td>
<td>.60</td>
<td>1.00</td>
<td>.60</td>
<td>.60</td>
<td>1.00</td>
<td>.46</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
</tr>
<tr>
<td>4,6</td>
<td>.96</td>
<td>.64</td>
<td>.86</td>
<td>1.00</td>
<td>.96</td>
<td>.64</td>
<td>.13</td>
<td>.98</td>
<td>.97</td>
<td>.13</td>
<td>.98</td>
<td>.77</td>
<td>.64</td>
<td>.97</td>
<td>.79</td>
</tr>
<tr>
<td>6,4</td>
<td>1.00</td>
<td>.36</td>
<td>1.00</td>
<td>.36</td>
<td>1.00</td>
<td>.36</td>
<td>.24</td>
<td>.61</td>
<td>1.00</td>
<td>.29</td>
<td>.79</td>
<td>.71</td>
<td>.36</td>
<td>.79</td>
<td>.84</td>
</tr>
<tr>
<td>Average</td>
<td>.96</td>
<td>.82</td>
<td>.97</td>
<td>.86</td>
<td>.96</td>
<td>.82</td>
<td>.38</td>
<td>.90</td>
<td>.96</td>
<td>.88</td>
<td>.92</td>
<td>.67</td>
<td>.82</td>
<td>.96</td>
<td>.88</td>
</tr>
</tbody>
</table>
5.4.3 Aligning with Inferred Relations

In this section, we discuss the evaluation of patterns to align ontologies using inferred relations. The performance of patterns to align ontologies using relations from within the original ontologies is shown in Table 5.7. Inferred relation Pattern 1 achieved high precision, but low recall in pairs 1,4 and 4,1. Analysis showed that some synonym pairs are expected but not returned by WordNet. For example, we expected \{ Location, Place \} to be returned as synonym pairs, but they were not returned by WordNet. We also used OpenCyc to see if we could improve recall in these cases, but it did not change the results. We found that this is because OpenCyc does not typically perform better than WordNet in identifying synonymy relations. However, overall, performance of pattern 1 was very promising. This is confirmed visually in the overall results for inferred relations, presented in the ROC scatter diagram in Figure 5.16. Pattern 1 has a fairly high true positive rate and low false positive rate.

The precision of Inferred Relations Pattern 2 is not as high as that of Pattern 1, as shown in Table 5.7 and Figure 5.16. There is a great incidence of low recall in the pairs. It was determined that hyponym pairs are not being provided by WordNet as expected. In some cases, this is because compound words are used to name classes in the ontologies. For example, the concept BSStudent is not recognized as a hyponym of Student in WordNet. This explains part of the poor recall performance. However, we did discover that quite a few hyponyms were not returned by WordNet.
Figure 5.16: ROC scatterplot showing the overall performance of inferred relations patterns.
Table 5.7: Evaluation of Inferred Relation Patterns

<table>
<thead>
<tr>
<th>Ontology Pair</th>
<th>Generic Relation r Pattern 1</th>
<th></th>
<th></th>
<th>Generic Relation r Pattern 2</th>
<th></th>
<th></th>
<th>Generic Relation r Pattern 3</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>WordNet</td>
<td>OpenCyc</td>
<td>Both</td>
<td>WordNet</td>
<td>OpenCyc</td>
<td>Both</td>
<td>WordNet</td>
<td>OpenCyc</td>
<td>Both</td>
</tr>
<tr>
<td>1,4</td>
<td>1.00</td>
<td>.57</td>
<td>1.00</td>
<td>.57</td>
<td>1.00</td>
<td>.57</td>
<td>.36</td>
<td>.57</td>
<td>.36</td>
</tr>
<tr>
<td>4,1</td>
<td>1.00</td>
<td>.77</td>
<td>1.00</td>
<td>.77</td>
<td>1.00</td>
<td>.89</td>
<td>1.00</td>
<td>.89</td>
<td>1.00</td>
</tr>
<tr>
<td>4,3</td>
<td>.90</td>
<td>1.00</td>
<td>.90</td>
<td>1.00</td>
<td>.61</td>
<td>.93</td>
<td>.64</td>
<td>.84</td>
<td>.65</td>
</tr>
<tr>
<td>4,5</td>
<td>.75</td>
<td>1.00</td>
<td>.75</td>
<td>1.00</td>
<td>.86</td>
<td>.97</td>
<td>.48</td>
<td>.33</td>
<td>.05</td>
</tr>
<tr>
<td>4,6</td>
<td>.82</td>
<td>1.00</td>
<td>.73</td>
<td>1.00</td>
<td>.57</td>
<td>.98</td>
<td>.42</td>
<td>.26</td>
<td>.12</td>
</tr>
<tr>
<td>Average</td>
<td>.89</td>
<td>.49</td>
<td>.94</td>
<td>.49</td>
<td>.89</td>
<td>.49</td>
<td>.78</td>
<td>.39</td>
<td>.83</td>
</tr>
</tbody>
</table>
When OpenCyc was added as a knowledge source, recall improved from 11% to 64% and 10% to 48% for ontology pairs 4.3 and 4.5, respectively. However, this is still a low recall value and we found that there are several reasons for this. First, there are semantic and pragmatic issues. Some of the ontologies used in this alignment define *AcademicStaff* and *SupportStaff*, to be a subclass of *Person*. OpenCyc considers *Staff* to be a group. The authors of the ontologies could have used the term *AcademicStaffMember* for clarity. On the other hand, OpenCyc could be enhanced to recognize that *Staff* is sometimes used in practice to mean *StaffMember*. In general, it is clear that the performance of patterns that rely on cross-ontological relationship discovery using external knowledge sources will be impacted by the naming conventions, semantics and pragmatics of the ontological entities. This is a great example of the challenge of automated alignment techniques. We will continue to explore ways to overcome these challenges. Despite the failings we saw in OpenCyc, we still found that it was very useful in providing hyponymic and hypernymic relations while WordNet performed slightly better for synonymic relations.

Inferred Relations Pattern 3 is the poorest performer with low recall when WordNet is used as the knowledge source. As has been seen throughout this Chapter, use of OpenCyc significantly improves performance. The overall error rate of alignment with inferred relations is shown in Figure 5.17. This chart shows that it is clear that while inferred relations have mostly very low false positive rate, false negative rates can be quite high. This suggests that improvement should be made to “loosen” the patterns, that is, relax the contraints of the patterns to perhaps tolerate a higher false alarm rate to gain a lower false negative rate. Also, the addition of more features to
Figure 5.17: Error rate of alignment with inferred relations.

detect inferred relations cross-ontology is likely to improve precision and reduce the false negative rate. A significant number of new features are added along with the machine learning technique. This enhancement is discussed in the next Chapter.

The overall performance of WordNet and OpenCyc are presented in ROC scatter diagrams in Figure 5.18 and Figure 5.19, respectively. The differences in performance of OpenCyc vs. WordNet become apparent in patterns that rely on hyponymy in particular and to some degree in patterns that rely on hypernymy. Clearly, OpenCyc delivers very accurate results, with very few false positives. In some cases, the true positive rate dips below 70%, but these are outliers from the typical performance. These results show that OpenCyc is a valuable knowledge source in ontology alignment. Further details on knowledge source performance can be found in [Stoutenburg and Kalita, 2008], [Stoutenburg et al., 2009].
Figure 5.18: ROC scatterplot showing the overall performance of WordNet in Inferred Relations patterns.

5.5 Conclusions

In this chapter, we presented a set of patterns that are designed to extend the relationships that can be derived between ontological entities in the ontology alignment process. In particular, we present patterns to learn non equivalence relationships, including subclass, superclass and inferred relationships, the latter of which aligns ontologies using relationships expressed in the original ontologies. The patterns use the semantics of OWL and external knowledge sources, including the upper ontology knowledge OpenCyc and online linguistic resource WordNet. Performance of the algorithms to acquire subclass and superclass relations are promising, with fairly high precision and recall. This is especially promising since the algorithms are simple
and easy to implement. Primary sources of error include unexpected relations from WordNet and missing relations in WordNet. Performance of alignment using inferred relations is encouraging, though the evaluation showed the resulting alignments were less accurate than the others. However, inferring relations across ontologies is a powerful approach since relationships unique to each aligned pair are acquired. Therefore, we might expect the alignment process to be more challenging.

In addition, we found that patterns that rely on synonymic relations cross-ontology work fairly well using WordNet as the source of online knowledge. We found that OpenCyc used in conjunction with WordNet may improve performance in patterns that rely on synonymy cross-ontology; but degrades performance in those
patterns that rely on hyponymy and hypernymy cross-ontology. We found that the performance of patterns that rely on cross-ontological relationship discovery using external knowledge sources will be impacted by the naming conventions of the ontological entities. We found that OpenCyc discovers a richer set of relationships. However, there were some challenges that were not overcome, including semantic and pragmatic challenges of class naming in the ontologies.

We submit that the strong performance of OpenCyc suggests a number of research extensions. First, it might be useful to extend this work to include ResearchCyc. ResearchCyc\(^3\) is an upper ontology that consists of a large knowledge base of commonsense knowledge. ResearchCyc contains significantly more concepts and assertions than OpenCyc and is therefore likely to enrich the information that can be used in alignment. (It was not used in this research because the Cyc Foundation does not grant licenses to individual users, only to institutions.) In addition, since OpenCyc is a very reliable source of subclass and superclass relations, it may be useful as a reasoner over biomedical data. It may be interesting to pursue transformation of biomedical knowledge into a biomedical version, perhaps a “BioCyc” of sorts. This idea is discussed further in Section 8.3.1.1.

Also, as part of this work, we contribute a reference alignment in the Academic domain that can be used in future evaluations of ontology alignment techniques. This is particularly useful since no such data sets exist for alignment using subclass, superclass and inferred relations. This data set will be available on the Web; and we plan to contribute it to the Ontology Alignment Evaluation Initiative (OAEI) in

\(^3\)http://research.cyc.com/
To improve the performance of alignment, we decided to investigate the use of machine learning. We decided to model the ontology alignment problem in a Support Vector Machine to determine if alignment performance could be improved. This work is discussed in the next Chapter.
Chapter 6

Learning to Align with Non Equivalence Relations

In this Chapter, we provide a brief description of Support Vector Machine technology and show how the ontology alignment problem can be modeled in a Support Vector Machine. We describe in some detail the feature extractors that are developed to support alignment using subclass, superclass and inferred relations. We describe the methodology used to align using SVMs. We evaluate the approach using ontologies from the Academic domain and demonstrate that ontology alignment accuracy is significantly enhanced using Support Vector Machines. Once we establish that SVMs improve alignment performance, we apply our approach to align ontologies in the Biomedical domain. We evaluate the approach and present an architecture for ontology alignment.

6.1 Support Vector Machines

Support Vector Machine (SVM) technology is a supervised machine learning technique based on statistical learning theory, first defined by Vapnik in the 1970s and described in [Vapnik, 1998]. SVMs support classification, regression and outlier detection. In SVMs, data in an n-dimensional space is classified by construction of hyperplanes in the space that maximize the margin separating each of n data sets.
The goal of the SVM is to construct a model that can be used to predict membership in one of the \( n \) classes based on a set of features of the data [Hsu et al., 2008]. Each datum is modeled as a pair composed of a feature vector and an indicator of class membership. To model training data, we define each training instance as a pair \((x_i, y_i), i = 1,...,l\) where \( x_i \) is a feature vector of \( j \) features and \( y_i \in 1, -1 \). The SVM seeks to find the optimal hyperplane that maximizes the margin between classes of data, as shown in Figure 6.1(a) (adapted from [Habib, 2008]). The problem is to maximize the distance between hyperplanes while still separating the data; that is, to select \( w \) and \( b \) to maximize \( f(x) = w^T x_i + b \) such that \( w^T x_i + b = 1 \) for data that are support vectors and \( w^T x_i + b > 1 \) for all other data points.

In the linearly separable case, we seek to (1) find two hyperplanes such that there are no points between them and (2) maximize the distance between them. This case is shown in Figure 6.1(b). The distance between hyperplanes is \( \frac{2}{\| w \|} \) therefore, the task is to minimize \( \| w \| \). In practice, the equation is altered to avoid square root functions (required for the vector norm of \( w \)), so the task becomes to find \( w \) and \( b \) that minimize \( \frac{\| w \|^2}{2} \) subject to \( y_i(w^T x_i + b) \geq 1, \forall x \) [Vapnik, 1998]. Note that \( y_i \) is a constant.

In some cases, it is not possible to find a linear hyperplane to separate classes. To solve these cases, errors are tolerated; these errors are represented by \( \xi \). As shown in Figure 6.1(c), from [Habib, 2008], data points may fall into three areas: (1) points falling outside the margin that are correctly classified, \( \xi_i = 0 \); (2) points falling outside the margin that are not correctly classified, \( \xi_i > 1 \); and (3) points falling within the margin that are still correctly classified \( 0 < \xi_i < 1 \). The optimization problem then
becomes to maximize the margin while minimizing error. Mathematically, this task can be stated as \[ \frac{\|w\|^2}{2} + C \sum \xi_i^2 \] subject to \[ y_i (w^T x_i + b) \geq 1 - \xi_i, \xi_i > 0, \forall i \] [Vapnik, 1998]. In this work, we found that the ontology alignment problem is not linearly separable; therefore, we had to perform parameter optimization to select C for optimal performance. Parameter optimization was part of the procedure to train the SVM; in fact, we found it to be an essential part of the process, as will be described later in this Chapter.

In the non linearly separable case, it is necessary to map the input space into a higher dimensional space. This is done through the use of the kernel function, referred
to as $\phi$. There are several types of kernel functions that can be used; in this work, we used the radial basis function, expressed as follows: $K(x_i, x_j) = \exp^{-\gamma \|x_i - x_j\|^2}$, $\gamma > 0$. Learning the optimal value for $\gamma$ became part of the procedure for training the SVM, along with finding the optimal value for $C$ as discussed above. More details on the optimization problem solved by SVMs and associated kernel functions can be found in [Vapnik, 1998], [Hsu et al., 2008], [Boser et al., 1992], [Cortes and Vapnick, 1995] and [Habib, 2008].

Support Vector Machines provide a valuable method for classification since they can handle generalized problems and high-dimensional input data [Habib, 2008]. SVMs do not suffer from local minima problems, as do neural networks and decision trees, because the training involved results in convex optimization [Smith et al., 2002]. In addition, SVMs produce stable and reproducible results, however, they can suffer some slow training, particularly when using non linear kernels and large input [Habib, 2008].

6.2 Modeling Ontology Alignment in a Support Vector Machine

Though the patterns described in Chapter 5 yielded promising results, we wanted to introduce a more flexible approach in which new features that suggest a correspondence could be added quickly and easily. Therefore, the addition of a machine learning component was an obvious choice. We decided to model ontology alignment using a Support Vector Machine to see if we could improve the results
discussed in Chapter 5. In addition, ontology alignment can be viewed as a binary classification problem so SVMs seemed a fit to the problem space. To our knowledge, ontology alignment approaches developed to date have not attempted to use this type of learning technique to improve alignment performance, so we were also hoping to explore an unchartered area in ontology alignment.

We modeled ontology alignment in an SVM as follows. Consider a set of ontologies we collect from the Web: \( O = o_1, o_2, \ldots, o_n \). We operate over one ontology pair at a time. Let an ontology pair be \((o_i, o_j)\). Let the set of \( k \) classes in ontology \( o_i \) be denoted by \( c_{i1}, c_{i2}, \ldots, c_{ik} \). Let the set of \( l \) classes in ontology \( o_j \) be denoted by \( c_{j1}, c_{j2}, \ldots, c_{jl} \). Let \( R \) be the set of relations to acquire cross-ontology; that is, \( R = \{ \text{subclass, superclass, inferred relations} \} \). Ontology alignment is the process of evaluating each class pair \((c_{ik}, c_{jl})\) to see if a relationship in \( R \) exists between them. We model this process in an SVM by letting each object in the SVM represent a class pair. We model a set of features that exist between each class pair in order to evaluate whether \( R \) exists between the pair. One set of features is defined per relationship. A simple notional model is shown in Figure 6.2. In this figure, each datum in the model represents a class pair. We model features as conditions that exist between each class pair. This is an important distinction in our use of SVMs, unique to the ontology alignment problem and we believe, a unique contribution. The features used to identify alignment are described further in the next two sections. The SVM predicts whether or not a relationship \( R \) exists between each pair. In this case, we show a simple linearly separable case to assist in envisioning the problem, but in this work, the data is not linearly separable. Clearly, unique sets of features and unique SVM models will be
required for each relationship to be acquired cross-ontology.

In SVMs, kernel functions are applied to map training vectors to the higher dimensional space. SVMs can apply different kernel functions, including linear, polynomial, radial basis function and sigmoid [Hsu et al., 2008]. The radial basis function, or RBF, can handle the case when the relationship between class labels and attributes is non linear. The RBF is recommended since it can handle linear and high dimensional cases and suffers from less numerical difficulties [Hsu et al., 2008]. For these reasons, we used the radial basis function as the kernel function. In addition, we used the SVM as a binary classifier; that is, the SVM will predict whether or not a relationship exists between ontological classes.

Use of SVMs requires that each datum is represented as a vector of real numbers. So features that suggest alignment must be represented in numeric data. We scaled each feature value to a number in the range [0,1], as recommended in [Hsu et al., 2008].
Also, we found that it was essential to perform cross validation. Cross validation is the process of determining two parameters for the RBF kernel, $C$ and $\gamma$ in order to ensure that the SVM can optimally predict class membership, in our case that is, whether or not a correspondence exists between each class pair. We used the LibSVM framework [Chang and Lin, 2001] for this work. Note that the disadvantage of using Support Vector Machines for alignment is that the ontology alignment cannot be explained.

In the next section, we discuss in detail the features used to support ontology alignment in a Support Vector Machine.

### 6.2.1 Representation of Evidence Primitives and Features in a Support Vector Machine

In this Section, we will discuss the techniques used to represent evidence primitives and features in a Support Vector Machine. Recall from the discussion in Section 6.2 that we are operating over class pairs, each denoted by $(c_{ik}, c_{jl})$ where $c_{ik} \in$ ontology $o_i$ and $c_{jl} \in$ ontology $o_j$. Then, it would appear that we can represent features for each class pair in a straightforward manner, by specifying the $(c_{ik}, c_{jl})$ and the feature type. For example, consider the feature that represents a measure of string distance between two classes across ontologies, as defined in [Stoiilos et al., 2005]. We might represent this feature as a three-tuple, as follows. $<c_{ik}, c_{jl}, stringDistanceMeasure>$. However, in order to represent more complex features, such as the patterns presented in Chapter 5, we found that more information...
needed to be specified. For example, multiple knowledge sources were used to derive relationships between class pairs, including WordNet and OpenCyc and our experimental findings determined that performance of these knowledge sources was quite different under various circumstances. Therefore, it was essential to train the SVM with features that distinguished knowledge sources for optimal performance of the SVM. This would require representation of features using four tuples; for example, the four-tuple $<c_{ik}, c_{jl}, hyponymy, Wordnet>$ represents the feature that a directed hyponymic relation exists between $c_{ik}$ and $c_{jl}$ according to WordNet.

Now, in order for the SVM to perform, it is necessary to represent each pattern described in Chapter 5 as a single feature. Each pattern requires that multiple conditions hold, so it was important to train the SVM using the complete pattern, since individual conditions may not meaningful. For example, simply the existence of superclasses of $c_{ik}$ means very little unless used in conjunction with other information. For example, if a superclass of $c_{ik}$ has a synonym $c_{jl}$, then a subclass relation may exist cross-ontology between $c_{ik}$ and $c_{jl}$. In addition, in most patterns, it is necessary that an intermediate class be present. For example, consider again Subclass Pattern 1, revisited in Figure 6.3. In this case, we seek to model a feature between class pair $(c_{ik}, c_{jl})$. To do that, it is necessary that a synonymy relation exists between the second member of the class pair, $c_{jl}$, and some intermediate class in $a_i$, which we’ll call $c_{im}$. It was also necessary that a subclass or hyponymy relation exist between the first member of the class pair, $c_{ik}$ and the intermediate class $c_{im}$. Therefore, the relationships that must exist are not always between the class pair $(c_{ik}, c_{jl})$; some intermediate class may be required. We found that this is often the case with the
Figure 6.3: Subclass Pattern 1 revisited. Note that to model a feature to discover a relation between \((c_{ik}, c_{jl})\), multiple conditions must hold and an intermediate class \(c_{im}\) must be discovered.

patterns described in Chapter 5. This means that multiple four-tuples will be required to express a single feature. For this reason, we distinguish between evidence primitives and features as follows.

In order to model features that consist of multiple conditions, we first model an evidence primitive as follows. We express each evidence primitive as a four-tuple \(<c_1, c_2, r, s>\), such that:

- \(c_1 \in o_i \cup o_j\)
- \(c_2 \in o_i \cup o_j\)
- \(r\) a relation not necessarily \(\in o_i \cup o_j\)
- \(s\) is the source of this feature

A feature consists of one or more evidence primitives. This definition allows
that \( c_1 \) is not necessarily \( \in o_i \) and \( c_2 \) is not necessarily \( \in o_j \) as required to represent the patterns described in Chapter 5. For example, consider again the feature that represents a measure of string distance between two classes. This is a simple example of a feature that has one primitive. We represent this feature as a single evidence primitive \(< c_{ik}, c_{jl}, stringDistanceMeasure, Stoilos >\). We represent Subclass Pattern 1 in the feature extractor as the presence of two evidence primitives:

- \(< c_{im}, c_{jl}, synonymy, WordNet >\)
- \(< c_{jl}, c_{im}, hyponymy, WordNet >\)

When variations on the two different knowledge sources are considered, this one pattern can be represented in four ways, as four features, as follows.

- \(< c_{im}, c_{jl}, synonymy, WordNet >\) and \(< c_{jl}, c_{im}, hyponymy, WordNet >\)
- \(< c_{im}, c_{jl}, synonymy, WordNet >\) and \(< c_{jl}, c_{im}, hyponymy, OpenCyc >\)
- \(< c_{im}, c_{jl}, synonymy, OpenCyc >\) and \(< c_{jl}, c_{im}, hyponymy, WordNet >\)
- \(< c_{im}, c_{jl}, synonymy, OpenCyc >\) and \(< c_{jl}, c_{im}, hyponymy, OpenCyc >\)

### 6.2.2 Features to Support Ontology Alignment

There are five main types of features used in this work to identify correspondences: pattern-based (which use semantics, structure and external data sources), string-based, linguistic-based, bag-of-word-based and extrinsic-based (which include syntactic and semantic knowledge). We discuss each of these types below. We selected features that are domain-independent so that the models can be used broadly.
There were over 50 features defined over all relations types in the SVM, with over 20 used per relationship type. Many of the features were reused. There was a heavy reliance on pattern- and linguistic-based approaches as well as use of the OpenCyc upper ontology and WordNet.

First, we discuss the string and linguistic-based features in our approach. Some of the string-based features include simple comparisons such as, detection that class names in a pair end with the same $n$ words. Another feature detects if words end in the same substring. We also use the string metric by Stoilos et al. [Stoilos et al., 2005], which is designed to be used in ontology alignment. This approach is motivated by the substring metric which computes scores based on the largest common substring between strings. This metric has been shown to enhance performance in ontology alignment techniques that seek to acquire equivalence. We decided to use it here to determine its effectiveness in aligning using other relationships. Some of the linguistic-based features include detection of meaningful prefixes in strings, such as sub-, super- and others that might suggest a subclass relationship, for example.

We also used bag-of-word processing to extract features of class pairs. This was particularly useful in ontologies in which class names consisted of multiple words, typical for biomedical ontologies. For example, we scored the number of synonyms, hyponyms and hypernyms in each class pair. Interestingly, we used an additional string-based technique to obtain inferred relationships cross-ontology. We performed a string distance metric that operated over the relationship name from the original ontology and each of the class names in different combination. This was designed to identify relationships across the biomedical ontologies, such as (PositiveRegulationOf-
PlateletActivation positivelyRegulates PlateletActivation."

Finally, extrinsic data sources were used in our feature extractors. We detect syntactic and semantic relationships between the class names using online knowledge sources, in particular, WordNet and OpenCyc. WordNet and OpenCyc are consulted to identify direct relationships between class pairs, such as synonymy, hyponymy and hypernymy. The sources are used to identify direct relationships between classes in a pair but are also used in the patterns described earlier in this section.

The complete set of evidence primitives used to align using subclass, superclass and inferred relations is provided in Appendix C.

6.2.3 Architecture for Ontology Alignment

The architecture used in this work for ontology alignment is shown in Figure 6.4. The architecture consists of a Feature Extractor and Support Vector Machine. The Feature Extractor identifies the features that exist between each class pair using the feature types described in Section 6.2.2. The Feature Extractor applies multiple techniques, including string-, linguistic-, structure-, semantic- and pattern-based approaches to derive features. In addition, the Feature Extractor makes use of the semantics of OWL and consults external knowledge sources including OpenCyc and Wordnet. The Feature Extractor represents each class pair as a vector of features identified. The SVM is then applied. The SVM utilizes training data to optimize the parameters $C$ and $\gamma$ and to construct a model. The SVM model is used to predict class membership between each class pair; that is, the SVM predicts whether a par-
ticular relationship exists between each class pair based on the features extracted for each pair. Of course, a unique Feature Extractor and SVM model were developed for each relationship to be acquired cross-ontology. The alignment result is expressed as an XML 4-tuple, including each class from the class pair, the relationship, and a relationship indicator indicating whether the relationship exists between the pair. In this work, we predicted discrete values, so the relationship indicator is $\in [0, 1]$.

6.3 Evaluation

To validate our approach, we employed the following methodology. First, we aligned ontologies from the Academic domain using the SVM. We compared the precision and recall of the SVM with the alignment results obtained using patterns, described in Chapter 5. Next, we aligned ontologies from the Biomedical domain
using the SVM approach and measured precision and recall. We describe each step in this process below.

In our evaluation using the Academic domain, we used the same reference alignment as in the previous experiments. For the Biomedical domain, we developed a reference alignment of correspondences semi-automatically; the reference alignments were manually specified with the help of biology subject matter experts, but the process was aided with software for efficiency. We selected pairs of ontologies to map, as before; the alignment is obviously directional, so different evaluations were done with ontology pairs (1,4) and (4,1), or (go,mp) and (mp,go), for example. Pairs were selected to ensure that we test as many unique combinations of alignments as possible. Some pairs were not tested as they either yielded an empty reference alignment or yielded similar alignments as in other pairs; this depended on the characteristics of the ontologies and the pattern applied.

To align the biomedical ontologies, a subset of alignment examples were chosen and a reference alignment was developed. The initial test cases covered include the following subject areas.

1. Platelet Activation

2. Mannose Binding

3. Immune System

4. Phenylalanine Conversion

5. Bone Remodeling
6. Bone Marrow

7. Osteoblast Differentiation

8. Osteoclast Differentiation

9. Behavior

10. Circadian Rhythym

The biomedical reference alignment is provided in Appendix B. For subclass and superclass, the size of the random data sets was 330; for inferred relations, random data set size was 1433.

Next, we applied the SVM for classification. Training data was assembled in the form:

\[ D = \{(x_i,c_i)|x_i \in \mathbb{R}, c_i \in \{-1,1\}, i = 1...n. \]  

\( c_i \in [1,-1] \) indicates the class to which the point belongs. That is, \( c_i \) indicates whether a specified relationship exists or not between a class pair. Each \( x_i \) is a \( p \)-dimensional vector representing the features of the ontology class pair. Each feature value was scaled to [0,1]. For most features, this required no modifications since the feature evaluation yields discrete results. For some features, such as string distance metrics, thresholds were used to scale the feature values. For example, if the Stoilos string distance was \( \geq 0.5 \), then the feature was scaled to 1.

To train the SVM, we built feature extractors which, given an ontology class pair \( c_{ik} \) and \( c_{jl} \), extracts the features defined in Section 6.2.2. One feature extractor was built for each relationship to be acquired. We used the feature extractor to construct
Figure 6.5: Methodology to train the Support Vector Machine.

a large set of data with known outcomes, as defined in the reference alignment. We randomly divided the reference alignment into training and test data (2/3 and 1/3, respectively). We used training data to train the SVM and used the resulting model to predict alignment in the test data. This was performed three times and the average performance was calculated over all random sets of data data. We used tools to optimize the parameters for the SVM, as described in [Hsu et al., 2008] and [Boser et al., 1992]. We wrote custom code to calculate precision and recall since LibSVM only reports accuracy. This methodology is shown in Figure 6.5. Different SVM models were built for each relationship to be acquired cross-ontology.
6.4 Experimental Results

First, we present the performance results of ontology alignment in the Academic domain under a variety of conditions. Recall that the first set of tests aligned ontologies in the Academic domain using the patterns described in Chapter 5. During alignment, we noticed varying performance when using WordNet and OpenCyc. Therefore, we ran multiple tests using WordNet as the only knowledge source, OpenCyc as the only knowledge source and WordNet and OpenCyc as knowledge sources. These results are summarized in Table 6.1 for comparison with the performance of the SVM. As can be seen in Table 6.1, alignment performance improved significantly. Comparing the average precision and recall in the Academic domain, the SVM improved performance for alignment using subclass relations from 72% precision to 99%; and from 88% recall to 94%. Performance of alignment using superclass also significantly improved, from 76% and 87% to 92% and 93% respectively. However, it should be noted that alignment using patterns that used OpenCyc as the sole external knowledge source did outperform the SVM, with slightly higher precision. Precision for inferred relations also improved, rising from 76% to 96%, and recall improved slightly from 53% to 56%. Note that the SVM uses both WordNet and OpenCyc as knowledge sources but identifies information from the knowledge sources as discrete features; thus, the SVM performs very well, despite the limitations of the knowledge sources previously discussed. In addition, the extra features implemented in the SVM are obviously effective.

Once we determined that the performance of alignment was improved with the
Table 6.1: Performance of alignment of academic ontologies, under a variety of test conditions.

<table>
<thead>
<tr>
<th>Test Condition</th>
<th>Subclass</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Prec</td>
<td>Rec</td>
<td>F</td>
<td>Prec</td>
<td>Rec</td>
<td>F</td>
<td>Prec</td>
<td>Rec</td>
<td>F</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Overall Average</td>
<td>.72</td>
<td>.88</td>
<td>.80</td>
<td>.76</td>
<td>.87</td>
<td>.81</td>
<td>.76</td>
<td>.53</td>
<td>.63</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OpenCyc is sole knowledge source</td>
<td>.97</td>
<td>.88</td>
<td>.92</td>
<td>.96</td>
<td>.88</td>
<td>.92</td>
<td>.93</td>
<td>.68</td>
<td>.79</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>WordNet is sole knowledge source</td>
<td>.64</td>
<td>.88</td>
<td>.74</td>
<td>.66</td>
<td>.84</td>
<td>.74</td>
<td>.69</td>
<td>.54</td>
<td>.61</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OpenCyc is knowledge source in patterns that rely on hyponymy, hypernymy</td>
<td>.88</td>
<td>.87</td>
<td>.88</td>
<td>.96</td>
<td>.92</td>
<td>.94</td>
<td>.84</td>
<td>.60</td>
<td>.70</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>WordNet is sole knowledge source in patterns that rely on synonymy</td>
<td>.91</td>
<td>.88</td>
<td>.90</td>
<td>.96</td>
<td>.82</td>
<td>.88</td>
<td>.89</td>
<td>.49</td>
<td>.63</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Average Performance using SVM</td>
<td>.99</td>
<td>.94</td>
<td>.96</td>
<td>.92</td>
<td>.93</td>
<td>.92</td>
<td>.96</td>
<td>.56</td>
<td>.70</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

SVM, we tested our approach in the Biomedical domain. The results of this testing are shown in Table 6.2. The SVM delivered high precision in the alignment of biomedical ontologies using subclass and superclass. Precision was less acceptable for inferred relations, at only 62%. Recall for alignment of biomedical ontologies was fairly low for all test conditions. Overall, alignment performance in the Biomedical domain is significantly poorer than that in the Academic domain. One reason for this is the complexity of terms in the biomedical ontologies. For example, consider terms such as Student and Conference that are typically present in Academic ontologies. These are commonsense terms contained in WordNet and OpenCyc. Features that rely on extrinsic knowledge will perform better when operating over class names that are commonsense terms. Consider typical class names from Biomedical ontologies, such as AbnormalEnzyme/CoEnzymeActivity and NegativeRegulationOfPlateletActivation. Using commonsense knowledge sources to obtain semantics of these types of
Table 6.2: Performance of SVM in alignment of the GO and MP ontologies.

<table>
<thead>
<tr>
<th>Test Condition</th>
<th>Subclass</th>
<th></th>
<th></th>
<th></th>
<th>Superclass</th>
<th></th>
<th></th>
<th></th>
<th>Inferred relations</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Prec</td>
<td>Rec</td>
<td>F</td>
<td></td>
<td>Prec</td>
<td>Rec</td>
<td>F</td>
<td></td>
<td>Prec</td>
<td>Rec</td>
<td>F</td>
</tr>
<tr>
<td>Biomedical Test 1</td>
<td>.81</td>
<td>.62</td>
<td>.70</td>
<td></td>
<td>.79</td>
<td>.55</td>
<td>.65</td>
<td></td>
<td>.66</td>
<td>.66</td>
<td>.66</td>
</tr>
<tr>
<td>Biomedical Test 2</td>
<td>1.00</td>
<td>.53</td>
<td>.69</td>
<td></td>
<td>.80</td>
<td>.42</td>
<td>.55</td>
<td></td>
<td>.65</td>
<td>.37</td>
<td>.47</td>
</tr>
<tr>
<td>Biomedical Test 3</td>
<td>.75</td>
<td>.47</td>
<td>.58</td>
<td></td>
<td>.92</td>
<td>.67</td>
<td>.77</td>
<td></td>
<td>.56</td>
<td>.54</td>
<td>.55</td>
</tr>
<tr>
<td>Average</td>
<td>.93</td>
<td>.54</td>
<td>.68</td>
<td></td>
<td>.84</td>
<td>.55</td>
<td>.66</td>
<td></td>
<td>.62</td>
<td>.52</td>
<td>.57</td>
</tr>
</tbody>
</table>

terms will not yield good results, typically, even when each word in the class name is considered individually. Even more commonly used terms such as *Skeleton*, are not well represented in WordNet and OpenCyc. For example, similarity between *Skeleton* and *Bone* are not returned by these sources, thus impacting overall performance. We considered using external sources from the biomedical domain, but so far have resisted this, since we seek to build a capability that is domain independent if at all possible.

In addition, some of the biomedical terms are very similar, both lexically and linguistically, but have no relation according to how the classes are defined by subject matter experts. For example, consider the terms *RegulationOfBoneResorption* and *AbnormalBoneResorption*. These terms score low string distances and are composed of very similar terms. However, there is no hyponymic nor hypernymic relations between the two. This type of error occurs frequently and significantly impacts the precision of alignment.

Another source of error is the fact that ontologies are manually generated and the associated reference alignment is manually generated. Sometimes, design decisions
in the original ontologies could be debated and the semantics and pragmatics of terms can be subtle. For example, one of the Academic ontologies specifies *Chair* as a subclass of *Person*. Clearly, the intent with the term *Chair* was actually *ChairPerson*, which is a subclass of *Person*. If the structure in the ontology is present to indicate this meaning, then our approach will detect the subtlety, but sometimes, the structure is not specified. In those cases, of course, our algorithms did not detect this alignment. At times, it is not clear what alignments should be made. For example, is *Conference* a subclass of *Meeting*? Is *Workshop* a *Place* or an *Event*? How can we detect the intended usage of the ontology? This lack of clarity occurred fairly infrequently in the Academic ontologies, but the impact in more complex domains such as biomedicine is more significant. These types of challenges are likely to remain a stumbling block to full automation of ontology alignment, at least in some domains.

Despite these challenges, we worked to identify additional features in the SVM to improve performance in the biomedical domain. We focused on one main area to improve: precision and recall performance in alignment for inferred relations. We analyzed the poor performance in this area and found that we could exploit the relationship and class names in additional ways. We developed additional features in the SVM that support linguistic analysis of the relationship name to the first class name in each class pair. For example, one new feature detects the linguistic pattern *(negative regulation of X, negatively regulates, X)*. We used similar patterns for *positively regulates* and *regulates*. The results of applying this set of heuristics are very promising, as shown in Table 6.3. Precision increased to 92% and recall to 73%. Note that this heuristic is specifically designed to optimize alignment of the GO and
MP ontologies. Therefore, it will not work in general. But, these results suggest that the approach of linguistic analysis of relationship and class names is likely to be a promising approach in general.

### 6.5 Conclusions

In this Chapter, we present an approach to enhance alignment performance by applying Support Vector Machine technology. We describe an original approach to modeling the ontology alignment problem in an SVM. We describe the representation and design of evidence primitives and features in this model. The features used in the SVM model are based on patterns, which use semantics, ontological structure and external knowledge sources. The SVM also classifies based on string, linguistic, syntactic and semantic characteristics that are present between each class pair. We also describe the architecture we used for alignment and the methodology used to train and use the SVM for identification of alignment.

We show that alignment is enhanced by the use of an SVM, significantly in some cases. We successfully aligned ontological concepts in multiple domains, including the Academic and Biomedical domains. Results of the approach are very
promising, with fairly high precision for subclass and superclass relations. Despite the varying performance of OpenCyc and WordNet, the SVM was able to perform well while utilizing both knowledge sources since features derived from each are discretely identified. Precision and recall were not as high for inferred relations, but we were able to show that specialized heuristics can be applied to improve performance. For example, we showed that linguistic processing of class and relation names can result in a significant enhancement in performance. However, these heuristics were optimized for performance in specific alignments, that is, the MP and GO alignment. In general, though, all of these results suggest that Support Vector Machines can be a valuable tool in advancing ontology alignment.

There remain some challenges in this area. The subtlety of semantics and usage of terms will remain a challenge, as described in the previous Section. Best practices in the development of ontologies and usage of standards, where applicable, would be a great approach to mitigating the challenge. For example, ontology authors could keep in mind a broader use of the knowledge, including use by machines. With that in mind, an ontology class could be named *ChairPerson* instead of *Chair*. Initiatives to develop standard taxonomies, which have been underway for years, will also assist in mitigating some of these issues. Additionally, meta data could play a role in describing the context of the knowledge in an ontology. Though ontologies, by nature, are intended to be self-describing, often in practice, are not. Therefore, the addition of meta data to ontologies could bring us closer to more effective methods of achieving self-describing ontologies. One such approach is presented in [Oro and Ruffolo, 2008]. In addition, ontology alignment could benefit from knowledge about the user, as well
as the role and intent of the user. More details on role-based alignment are described in Section 8.3.

In addition, we contribute a reference alignment in the biomedical domain that can be used in future evaluations of ontology alignment techniques. This data set will be available on the Web; and we plan to contribute it to the Ontology Alignment Evaluation Initiative (OAEI) in the future.

In the future, we hope to demonstrate new applications of ontology alignment in the biomedical domain that could facilitate in hypothesis generation, streamline the data analysis process and bring to light relationships that might be overlooked if this analysis was done manually. More details on these ideas are described in Section 8.3.
Chapter 7

Scaling Ontology Processing, Reasoning and Alignment

One of the goals of this research was to apply our approach to align large-scale biomedical ontologies. At the onset of large-scale testing, however, we encountered significant challenges in opening large scale ontologies for processing. We found similar challenges even when using open source tools that are widely used in the semantic community, such as the Protege tool for editing and managing ontologies. In addition, when feature extraction was applied in the alignment process, the application ran initially for up to three days when processing fairly small numbers of ontology pairs. Therefore, it became apparent that investigating scalability would be an essential part of this research.

To date, research into developing scalable infrastructures for processing and reasoning over ontologies has been in two main areas: first, developing a representation approach that combines semantic richness (i.e., expressiveness) with scalable performance; and second, harnessing the power and scalability of database systems to represent ontologies [Hepp et al., 2008]. Solutions to scaling ontology alignment are only starting to emerge. In fact, scalability continues to be one of the grand challenges of ontology construction and use [Hepp et al., 2008] and scaling ontology alignment in particular remains a research area [Giunchiglia et al., 2009].
In this Chapter, we discuss scalability issues in the area of processing, reasoning, and aligning ontologies with large sets of classes and properties. We survey related work in the area and discuss a number of approaches that we developed to improve scalability and performance in ontological processing and reasoning. We present an alignment algorithm that bounds the processing by selecting optimal subtrees to align. We show that our work improves the efficiency of alignment without significant reduction in accuracy. We discuss next steps in this important area. We omit a discussion on ontology alignment when dealing with large sets of instances since our work has focused on alignment of biomedical ontologies that are composed of large sets of classes and properties.

7.1 Scalability in Ontology Processing and Reasoning

In this work, it was necessary to process ontologies in order to reason over them. Reasoning was required to obtain information about ontologies for use in alignment. The ontologies were queried for information such as classes and their associated labels, relationships between classes that indicate structure and semantics, superclasses and subclasses of each respective class, etc. Ontology processing was performed using the open source Protege OWL API\(^1\), which offers a rich set of features over native OWL ontologies. Once we started testing with large ontologies, however, we encountered difficulties in reading in and processing large ontologies. For example, the GO ontology in native OWL is nearly 57GB in size and contains well over 27,000

\(^1\)http://protege.stanford.edu/plugins/owl/api/
classes. We were unable to open the ontology for processing. Even more moderately sized ontologies, such as the Cellular Component ontology, a subset of GO, would not open, with over 2300 classes and a native OWL file size of 3.5 GB. Note that initial testing was performed on a virtual machine running on a quad-core 2GHz Intel Xeon processor, with 2GB of RAM, and a 15K SAS hard drive. Challenges opening the files continued after building a 64-bit machine and optimizing Java heap size. To give an idea of what is meant by “large-scale” ontologies, we present sizes of some of the largest biomedical ontologies in Table 7.1, as well as the sizes of ontologies used in the Ontology Alignment Evaluation Initiative (OAEI) in 2007 and 2008.

To investigate the issue further, we attempted to use open source and industry-developed tools to determine if they could process the large ontologies. We had difficulty opening these ontologies with the Protege ontology editor even with maximum

---

Table 7.1: Size comparison of large biomedical ontologies and ontologies used in the Ontology Alignment Evaluation Initiative (OAEI) 2008.

| Ontology                          | |OWL Classes| | Approx. OWL File Size |
|-----------------------------------|-----------------|-----------------|
| OAEI Anatomy: Adult Mouse         | 2744             | 1.4 MB          |
| OAEI Anatomy: NCI Thesaurus       | 3304             | 2.9 MB          |
| OAEI Food : AGROVOC               | 10000            | 30 MB           |
| OAEI Food : ASFA                  | 2500             | 7.8 MB          |
| OAEI Library: GTT                 | 35194            | n/a             |
| OAEI Library: Brinkman            | 5221             | n/a             |
| Gene Ontology                     | 27598            | 57 MB           |
| Mammalian Phenotype Ontology      | 3000             | 11 MB           |

---

Java heap space on a 64-bit machine. We tested TopBraid Composer\(^3\) to determine if it could open the GO and MP owl ontologies. TopBraid did successfully open both ontologies without additions to heap size but processing was very slow. TopBraid software is proprietary, so the details of design are not available; however, it appears that TopBraid stores RDF and OWL triples in a database\(^4\). In addition, the GO browser online is using MySQL databases to support scaling to very large ontologies\(^5\). Pellet, an open source OWL DL reasoner, also uses a backend database to store triples.

We surveyed the literature and found that in fact, there are two major approaches to scaling ontology processing and reasoning. The first are tools that process and reason over native OWL files and the second are those tools that employ a database to store ontological triples. First, we discuss the latest approaches that reason over native OWL files. OWLIM [Kiryakov et al., 2005] is an approach that provides a scalable semantic repository for RDF and OWL Lite. OWLIM utilizes the Triple Reasoning and Rule Entailment Engine (TRREE)\(^6\) which performs forward chaining reasoning in main memory and caches the results for high performance query answering. It is claimed\(^7\) that a version of OWLIM, BigOWLIM 3.0, recently loaded 6.7 billion statements in OWL in 42 hours, a significant achievement. However, OWLIM is not open source nor free. AllegroGraph\(^8\) is another high performance

\(^3\)http://www.topquadrant.com/topbraid/composer/index.html
\(^4\)This is according to one of its partner sites, http://agraph.franz.com/tbc/.
\(^5\)http://www.geneontology.org/GO.tools.annotation.shtml
\(^6\)http://www.ontotext.com/trree/
\(^7\)http://www.ontotext.com/owlim/big/index.html
\(^8\)http://www.franz.com/products/allegrograph
approach that operates over RDF triples. AllegroGraph claims to support the management of billions of triples. HStar [Chen et al., 2006] proposes a method whereby ontologies are organized in indexed hierarchies using memory-based hash map modules. Note that these approaches for large-scale processing and reasoning have only recently emerged since 2005. In addition, it is not clear that these approaches for managing large ontologies provide all the features needed in ontology alignment; for example, alignment requires access to direct subclasses, access to direct superclasses, etc.

A number of approaches have represented ontologies in relational database management systems. These approaches typically create a table for each class that contain instances for that class; for example, DLDB-OWL [Pan and Heffin, 2003] and Sesame on PostgreSQL [Boser et al., 2002]. DLDB-OWL employs database views to implement subsumption, while Sesame uses subtables. Minerva is an approach [Zhou et al., 2006] that implements a table for each property in which triples are stored. Schema alignment systems, such as Harmony [Mork et al., 2006] also leverage database management systems and use similar best practices in database schema design. In particular, Harmony constructs a single table for each property. Of course, for ontologies with thousands of properties, such as the Systematized NOmenclature of MEDicine (SNOMED)\(^9\), this approach would not be viable. It is important to note that Oracle has supported storage of RDF triples since version 10g\(^{10}\) and Oracle 11g supports RDF and OWL\(^{11}\). The cost of an Oracle solution and its lack of open source solutions

\(^9\)http://www.nlm.nih.gov/research/umls/Snomed/snomed_main.html
\(^{10}\)http://www.oracle.com/technology.tech/semantic_technologies/pdf/semantic_tech.rdf_wp.pdf
\(^{11}\)http://www.oracle.com/technology.tech/semantic_technologies/pdf/
has inhibited research; however, it is possible that large-scale industrial applications could leverage an Oracle database and associated functionality.

Approaches that operate over native OWL files typically reduce ontology load and update times but may not perform as quickly during query time. Database systems often provide query optimization features but typically take longer to load [Hepp et al., 2008]. In addition, native OWL file management approaches need to address issues already solved by the relational database community, such as transaction processing, access control, query optimization, etc. Therefore, in our view, it seems as if the most promising solution to managing very large scale ontologies is to represent the knowledge in a database. However, this requires that the reasoning capabilities of OWL must be implemented to operate over the database.

### 7.2 Scalability of Ontology Alignment

In 2009, scalability of ontology alignment remains a key challenge and approaches to address the challenge are only starting to emerge. As Giunchiglia et al. point out, “the number of possible correspondences between two ontologies grows quadratically with respect to the numbers of entities in the ontologies” [Giunchiglia et al., 2009]. In the Ontology Alignment Evaluation Initiative in 2007 and 2008, participants were encouraged to align large data sets, including the Anatomy, Food and Library ontologies [Euzenat et al., 2007], [Caracciolo et al., 2008]. However, only four of seventeen tools accomplished alignment tasks with the largest ontologies.
in OAEI 2007 and only three in OAEI 2008. Paulheim found that the very large biomedical ontologies, such as GO and NCI Thesaurus\textsuperscript{12} cannot be processed by top performing ontology alignment tools [Paulheim, 2008]. In addition, the Harmony Integration Workbench authors indicate that it is only after five years of development that they are starting to address the challenge of scalability in alignment. This is further evidence of the difficulty of the challenge.

### 7.2.1 Approaches to Scaling Alignment

Some early attempts at scaling ontology alignment include initial elimination of candidates prior to alignment. Typically, light-weight string-based matching has been applied to handle the task of candidate elimination in large-scale ontology alignment; for example, [Zhang and Bodenreider, 2007], [Kirsten et al., 2007]. The problem with these approaches is that they are comparing lexical qualities of ontological classes instead of semantic qualities. Thus, this approach risks reduced accuracy, in particular, reduced recall. In addition, these approaches, in particular, [Zhang and Bodenreider, 2007] and [Kirsten et al., 2007], use domain-specific knowledge sources which precludes these solutions from performing successful ontology alignment in other domains. Other work has adapted match algorithms to reduce the use of memory. Mork et al. showed how two large biomedical ontologies, the Foundational Model of Anatomy (FMA) and Galen\textsuperscript{13}, could be aligned by considering only direct subclasses and superclasses of each class to align. This work was successful in aligning

\textsuperscript{12}http://ncicb.nci.nih.gov/NCICB/infrastructure/cacore_overview/vocabulary

\textsuperscript{13}http://www.co-ode.org/galen/
large biomedical ontologies, but it is important to note that the ontologies were not expressed in OWL nor RDF; instead a frame-based system was used. In addition, a relational database system was used to store the ontology [Mork and Bernstein, 2004].

In [Ehrig, 2007], alignment efficiency is analyzed and critical elements for performance are identified; namely, number of candidates for comparison and costs for retrieving and comparing features. Ehrig describes how a set of heuristics can be applied to lower the cost of these critical elements. He transforms NOM to QOM through these techniques and successfully lowers the complexity of QOM to $O(n\log(n))$. However, increasing efficiency lowered the overall quality of the alignments [Ehrig, 2007]. Note that the heuristics used by Ehrig are tightly coupled to the alignment approach in NOM. This suggests that other heuristics could be developed for other alignment architectures. In this research, we worked to ensure that the heuristics remained domain-independent.

In OAEI 2007, only four of seventeen approaches aligned the largest set of ontologies, which include the Anatomy, Food and Library tracks [Euzenat et al., 2007]. One of the approaches, DSSIM, applied manual partitioning in order to process the large ontologies [Nagy et al., 2007]. Other tools, including RIMOM [Tang et al., 2006] and PRIOR+ [Mao and Peng, 2007] applied light-weight string-matching techniques as alternatives to full-scale alignment. An approach by Wang and Xu was presented at OAEI 2007 called Lily, but Lily did not align the largest ontologies until OAEI 2008. Wang and Xu make reference to an approach to scalability which they refer to as “generic ontology matching method based on semantic subgraphs,” but this
work has not yet been published [Wang and Xu, 2008a]. At the Ontology Matching workshop in 2008, Paulheim proposed two approaches to enhance the scalability of ontology alignment: overlapping partitions and optimized thresholding [Paulheim, 2008]. These are discussed in further detail in the next section.

### 7.2.2 Partitioning and Thresholding

Partitioning ontologies as an approach to improving scalability is risky, since most ontology alignment approaches rely on ontological structure to some degree. Performed naively, partitioning could lead to significant degradation in alignment accuracy [Paulheim, 2008]. One approach to avoiding this issue involves the creation of overlapping partitions. In this approach, the direct neighbors of each concept are included in the partition, thus creating overlapping partitions. This approach has shown significant improvement in alignment accuracy though there is significant cost in time complexity [Paulheim, 2008].

A divide-and-conquer strategy for ontology alignment is described in work by Hu in late 2008 [Hu et al., 2008]. In this approach, ontologies are partitioned into blocks by measuring structural proximity, with each block being labeled with RDF sentences [Zhang et al., 2007]14. Then, each block is mapped using entities matched beforehand and alignments are determined across blocks [Hu et al., 2008]. Hu reports that this approach reduced execution time of alignment, down to 12 minutes for the OAEI Anatomy track. By comparison, in OAEI 2008, DSSIM, RIMOM and

---

14An RDF Sentence is a unit of summarization used to summarize the content of an ontology [Zhang et al., 2007].
PRIOR+ report execution times of 4 hours, 75 minutes and 23 minutes, respectively. However, the execution time for the partitioning algorithm is not reported. The time complexity of Hu’s partitioning algorithm is $O(n^2)$; in addition, the partitioning algorithm uses a structural proximity matrix that is precomputed. Computation of this matrix is also expensive, $O(n^2)$. Therefore, use of partitioning simply moves the complexity of alignment to an a priori stage of processing. However this type of approach is still potentially valuable since it optimizes the alignment process, which is a step toward the possibility of near real-time alignment. The capability to align ontologies in real-time will be important as ontology alignments are used for dynamic Web service composition and mediation in the future. It is also important to note that the evaluation of the approach was performed using moderately sized ontologies, containing no more than 3500 classes each. Based on our research results, we believe this is likely because the partitioning process likely took days to run even on moderately sized ontologies. (Our work in this area will be discussed in detail later in this Chapter.) Furthermore, Hu’s structural proximity matrix does not consider the qualitative properties of the relations between ontological nodes in the graph. That is, each link in the graph is treated as if it were the same relation. Thus, if the relation between ontological concepts is disjoint, or some other relation that indicates semantic distance, then the structural proximity matrix approach breaks down in ontology alignment. Of course, any additional processing to determine the qualitative nature of the relations is likely to increase execution time of computing the proximity matrix.

Stuckenschmidt and Klein propose an algorithm for partitioning ontologies for
enhanced visualization and navigation [Stuckenschmidt and Klein, 2004]. This approach measures the degree of dependency between concepts, but does not maintain semantics of the partitions [Cuenca-Grau et al., 2007] Algorithms are proposed in [Noy and Musen, 2003] and [Seidenberg and Rector, 2006] to partition ontologies by traversing semantically close links (i.e., relationships between concepts) and produce stand-alone fragments. However, as pointed out in [Cuenca-Grau et al., 2007], these approaches do not characterize the logical properties of the extracted fragments. The notion of ontology modules and methods for extracting them are formally defined in [Cuenca-Grau et al., 2007]. These efforts partition ontologies based on the semantics of the terms. However, despite the strong accuracy of the results in [Cuenca-Grau et al., 2007] for example, these partitioning methods often result in modules (or partitions) that are still quite large, on the order of thousands of classes [Hu et al., 2008].

A second approach to enhancing scalability is the idea of thresholding. In this approach, a lower threshold to select pairs for alignment is used across partitioned ontologies. This approach was motivated by the observation that average confidence values for alignment are significantly lower for false positives than for true positives. Threshold levels were obtained by manually inspecting the optimal values across a set of data. It is claimed that this approach improves accuracy without significant increased cost in time complexity, over other partitioning methods [Paulheim, 2008].
7.3 Metrics for Large-Scale Ontology Alignment

Determining precision and recall in large-scale ontology alignment is difficult. Ideally, we would have access to a complete reference alignment with all “true” results; but to build such a set of data for large-scale alignments is manual, difficult and time-consuming, requiring large groups of domain experts. This is particularly true in alignments with non equivalence relations. Data sets with known results for alignment with subclass, superclass and inferred relations simply do not yet exist. Therefore, it is necessary to use metrics that estimate precision and recall. We decided to use a set of metrics proposed by [Kirsten et al., 2007] which applies rough approximations for precision and recall using relative quality measures. Kirsten identifies two primary accuracy approximation metrics: *match coverage*, designed to approximate recall, and *match ratio*, designed to approximate precision. Match coverage essentially measures how many classes are identified in the alignment. Let $c_{o1}$ be the set of all classes in ontology $o_1$ and let $c_{1-match}$ be the set of concepts in $o_1$ that appear in any alignment pair. Let $c_{o2}$ be the set of all classes in ontology $o_2$ and let $c_{2-match}$ be the set of concepts in $o_2$ that appear in any alignment pair. Then,

$$MatchCoverage_{o_1} = \frac{|c_{1-match}|}{|c_{o1}|} \quad \text{and} \quad MatchCoverage_{o_2} = \frac{|c_{2-match}|}{|c_{o2}|}$$

According to Kirsten, high match coverage indicates that a significant number of concepts were matched. For example, 90% match coverage would indicate that 90% of the classes are matched. This suggests high recall. Therefore, in general, we seek to attain match coverages that are high, above 60%.

To estimate precision, match ratio measures the ratio between the number of
correspondences and the number of matched concepts. The idea is that precision of a matched result is better if a single concept is matched to fewer concepts (preferably similar concepts) and not loosely matched [Kirsten et al., 2007]. To define match ratio, let $corr_{o1o2}$ be the correspondences found in the alignment process. Then,

$$MatchRatio_{o1} = \frac{|corr_{o1o2}|}{|c_{1\text{-match}}|} \text{ and } MatchRatio_{o2} = \frac{|corr_{o1o2}|}{|c_{2\text{-match}}|}$$

The combined match ratio is defined similar to f-score, as follows.

$$CombinedMatchRatio = \frac{2 * |corr_{o1o2}|}{|c_{1\text{-match}}| + |c_{2\text{-match}}|}$$

Match ratios that are too high indicate concepts mapped to many other concepts, a suggestion of low precision. Match ratios close to 1.0 indicate the highest precision. Kirsten states that match ratios between 2-9 indicate reasonable levels of precision.

We use these set of metrics to estimate precision and recall in our large-scale alignment results. We also perform random spot checks of the resultant alignments to strengthen confidence in the results.

### 7.4 A Scalable Infrastructure for Ontology Processing and Alignment

In this section, we describe the work we performed in order to process and align large scale biomedical ontologies.
7.4.1 Infrastructure for Ontology Processing

When work began to align large scale biomedical ontologies, we encountered difficulty immediately upon attempting to load large ontologies using the Protege API. Despite maximum heap size settings, we were unable to read in native OWL files using the Protege API. The Jena API is a subset of the Protege API; it has less features and seemed likely to be more efficient. We modified our application to use Jena but still were unable to open the larger ontologies for processing. It became clear that the ontologies needed to be loaded into a database in order for alignment to occur. We investigated existing tools that load ontologies into databases, but found that there were not sufficient APIs to support ontology alignment. We found that we needed direct access to the database. Therefore we designed a database schema to represent large scale ontologies. The logical design of the database is shown in Figure 7.1. The database consists of a main table Concept which stores all concepts in the ontology. We built one table per relationship to optimize query response time. In the biomedical ontologies of interest in this work, there were only 5 relation types: subclass, partOf, regulates, positivelyRegulates and negativelyRegulates. Each table stores the two concepts that share the relation represented by the table. As discussed earlier in this chapter, other approaches use this optimization technique [Pan and Heflin, 2003], [Boser et al., 2002], [Mork et al., 2006]. To ensure that our approach works in general, we included a table called Triple to store relations other than the 5 types found commonly in the Open Biomedical Ontologies. In order to store other types of relations in the Triple table, we also needed a Relation table to store the
Figure 7.1: Relational database design for scalable ontology processing and alignment. relation types. Note that we also include a table to store DirectSubclass relations; this became necessary for efficiency in ontology alignment and is discussed later in this Chapter. We implemented the database in MySQL 5.1\textsuperscript{15}.

To make use of this database design, we built a method to dynamically create a new database for each ontology to be processed using a generic SQL script. This method also loads ontological information into the database. It was also necessary to build a set of methods that mimic the reasoning capabilities of OWL. These methods provide functions such as, given a concept $c$, $getSuperclasses(c)$, $getSubclasses(c)$; also, given two concepts, $c, d$, $getDirectRelations(c, d)$, etc. Note that we omitted a function to compute the full transitive closure. While this is certainly a well-defined function that could be implemented, our alignment process does not require full transitive closure and transitive closure is very expensive computationally.

\textsuperscript{15}http://dev.mysql.com/downloads/mysql/5.1.html
We tested the method for database loading and recorded the times to load; the results are provided in Table 7.2. Initially, we were able to load the NCI Thesaurus (nci) and the Mouse Anatomy (ma) ontologies into the database in just seconds, but were unable to open others ontologies, such as the Cellular Component sub ontology of GO (go_cc). This result is indicated in Iteration 1. We worked to optimize the code where possible, including optimizing the interface to the database. For example, we optimized the SQL queries to the database. These results are shown as Iteration 2; this iteration improved our loading times somewhat, but we were still unable to load the larger subsets of the GO ontology. Finally, to reduce memory usage, we built an executable jar file and ran the application on a quad-core 2GHz Intel Xeon processor, with 3GB of RAM. We closed all other applications on the server. We also removed all other data in the database; this included a very large instance of Wikipedia. The results of this optimization is shown in Iteration 3. Using this configuration, we were able to load all ontologies into the database. The largest ontology, GO, took approximately 47 minutes to load.

7.4.2 Optimizing Feature Extraction in Ontology Alignment

Once we were able to successfully load the large biomedical ontologies into the database, we began testing alignment. We performed baseline measures of performance using the Mammalian Phenotype (mp) ontology and the Cellular Component subset of GO (go_cc). For just 100 class pairs, the feature extraction for subclass relations ran for over 6 hours. Clearly, optimizations were needed. The first optimization
Table 7.2: Loading time for large-scale biomedical ontologies.

<table>
<thead>
<tr>
<th>Ont.</th>
<th>Classes</th>
<th>Subclass relations</th>
<th>PartOf relations</th>
<th>Other relations</th>
<th>OWL File Size (MB)</th>
<th>Time to Load (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Iter 1</td>
</tr>
<tr>
<td>nci</td>
<td>3304</td>
<td>18356</td>
<td>3031</td>
<td>0</td>
<td>2.86</td>
<td>103</td>
</tr>
<tr>
<td>ma</td>
<td>2737</td>
<td>4632</td>
<td>2736</td>
<td>0</td>
<td>1.37</td>
<td>43</td>
</tr>
<tr>
<td>go_cc</td>
<td>2369</td>
<td>17207</td>
<td>2207</td>
<td>0</td>
<td>3.53</td>
<td>error</td>
</tr>
<tr>
<td>go_mf</td>
<td>9185</td>
<td>53155</td>
<td>3</td>
<td>0</td>
<td>23.10</td>
<td>error</td>
</tr>
<tr>
<td>go_bp</td>
<td>16062</td>
<td>182634</td>
<td>6558</td>
<td>7256</td>
<td>29.95</td>
<td></td>
</tr>
<tr>
<td>mp</td>
<td>6510</td>
<td>46401</td>
<td>0</td>
<td>0</td>
<td>10.43</td>
<td></td>
</tr>
<tr>
<td>go</td>
<td>27598</td>
<td>252860</td>
<td>8765</td>
<td>7246</td>
<td>56.49</td>
<td></td>
</tr>
</tbody>
</table>

we applied was to use direct subclasses and superclasses only in the feature extraction process. For example, in many of the features (described in Chapter 6), the extractor utilized all subclasses of a particular class; we modified this to use just direct subclasses. All features that rely upon superclasses or subclasses were optimized in this way. This approach does risk reduced accuracy, it should be noted. We tested this approach and found that performance was improved to some extent, but the results were disappointing. Next, we applied a second optimization in which we modified the interface to external knowledge sources. We created a single object for the WordNet and OpenCyc interface and passed the single object between methods, replacing the original approach of opening the sources when needed. This second optimization resulted in significant improvement in performance. In Figure 7.2, alignment performance is shown, comparing performance between the original algorithm and both optimizations with up to 100 pairs to align. Figure 7.3 shows alignment performance
with up to 1000 pairs to align.

![Alignment Performance Comparison for up to 100 Pairs](image)

Figure 7.2: Example of mapping classes cross-ontology using relationships within the original ontologies.

Despite this progress, these improvements represent enhancement in efficiency, not scalability. Therefore, we decided to pursue various methods to scale alignment. These efforts are discussed in the next sections.

### 7.4.3 Optimizing Ontology Alignment with Partitioning

Based on the results of Hu et al. [Hu et al., 2008], discussed in Section 7.2.2, we decided to explore solutions to partitioning using bottom-up agglomerative clustering techniques. Agglomerative clustering algorithms assign each node in a graph to its own cluster, then gradually merge clusters until some threshold is reached. Bottom-up clustering begins with leaf nodes in the graph and works up the nodes in the graph. The thought was that we could utilize Hu’s approach and explore ways to
potentially improve upon it by considering the qualitative nature of the relationships in the graph without significantly impacting performance. We based this decision on Hu’s report of significantly reduced execution time of alignment.

As Hu et al. did, we decided to base our approach on a classic agglomerative clustering algorithm, called the ROCK algorithm, introduced by Guha et al. in 1999 [Guha et al., 1999] as tailored by Hu. The partitioning algorithm is shown in Algorithm 1. Initially, each node (class) in the graph is assigned to its own partition (lines 12-15). Nodes to process are selected by depth in the graph; depth of nodes (classes) in the graph are tracked by measuring the distance from the root, as shown in lines 4-5. The proximity between nodes is initialized using a proximity measure, as shown in lines 8-10; this proximity between nodes is used to initialize the cohesiveness between partitions, lines 18-20. This algorithm works by maximizing the cohesiveness
between nodes within a partition and minimizing the cohesiveness between partitions.

We noted earlier in this Chapter that Hu used a method in which all links in the graph are treated equally; Hu does not consider the qualitative nature of the relations between nodes. Therefore, Hu’s approach would not work if the relations in the graph represent disjointness or other relations that indicate semantic distance between concepts. We introduce a unique cohesiveness formula which considers the qualitative nature of relations in the graph.

\[
\text{cohesiveness}(p_i, p_j) = w_1 f(p_i, p_j) + w_2 g(p_i, p_j) + w_3 h(p_i, p_j),
\]

where:

- \( p_i \) is a partition of ontology_i
- \( p_j \) is a partition of ontology_j
- \( f(p_i, p_j) = \) | commonsuperclasses(p_i, p_j) |
- \( g(p_i, p_j) = \begin{cases} 
1 & \text{if } \text{depth}(n_i) = \text{depth}(n_j) \\
\frac{1}{|\text{depth}(n_i) - \text{depth}(n_j)|} & \text{otherwise}
\end{cases} \)
- \( h(p_i, p_j) = \begin{cases} 
1, & \text{if } (\exists z)(r(p_i, p_z) \land r(p_j, p_z)) \lor (r(p_z, p_i) \land r(p_z, p_j)) \\
0, & \text{otherwise}
\end{cases} \)
- \( r \in \{\text{equivalence, similarity, subclass, superclass, partof}\} \)
- \( w_i \) are weights for each component

The function \( f(p_i, p_j) \) is designed to score based on the number of common superclasses of each partition pair. The function \( g(p_i, p_j) \) seeks to minimize the depth differences within a partition. The function \( h(p_i, p_j) \) is designed to score cohe-
siveness based on the existence of relations between nodes, namely, the relation set \{equivalence, similarity, subclass, superclass, partof\}.

The problem with this approach is that it is very expensive computationally, especially when the partitions are merged (lines 31-36). Here, the cohesiveness of each class within each partition must be recomputed. When we implemented this partitioning algorithm, we attempted to partition the NCI Thesaurus ontology. We found that the partitioning algorithm ran nearly as long as the alignment function, for a little over 2 days for this moderately sized ontology. It is interesting to note that Hu did not report on the time to partition; only the time to align [Hu et al., 2008]. So, while partitioning ontologies prior to alignment is a way to reduce alignment time, overall processing time is not reduced, in fact is significantly increased. The complexity of processing is merely pushed to an a priori process. Therefore, we abandoned this approach in search of a better answer.
Algorithm 1 partitionOntology($c_i \in C, \epsilon$)
1: Input: a set of concepts $C$ and a parameter $\epsilon$ that limits the maximum number of partitions, $\epsilon < |C|$  
2: Output: a set of partitions $P$  
3: //Initialization  
4: for each concept $c_i \in C$ do  
5: $d_i = 1 + |\text{superclasses}(c_i)|$  
6: end for  
7: for each concept $c_i \in C$ do  
8: for each concept $c_j \in C$ satisfying $i \neq j$ do  
9: proximity($c_i, c_j$) = $w_1 f(c_i, c_j) + w_2 g(c_i, c_j), w_3 h(c_i, c_j)$;  
10: end for  
11: end for  
12: for each concept $c_i \in C$ do  
13: $p_i = \text{newpartition}(c_i)$;  
14: $P = P \cup p_i$;  
15: end for  
16: for each partition $p_i \in P$ do  
17: depth($p_i$) = depth($c_i$);  
18: for each partition $p_j \in P$ satisfying $i \neq j$ do  
19: cohesiveness($p_i, p_j$) = proximity($c_i, c_j$);  
20: end for  
21: end for  
22: //Partition  
23: while not done do  
24: $p_s = \text{argmax}(\text{depth}(p_i)); //p_s \in P$  
25: $p_t = \text{argmax}(\text{cohesiveness}(p_i, p_j)); //p_t \in P, s \neq t$  
26: if $|p_s| + |p_t| > \epsilon$ or depth($p_s$) == 0 then  
27: done = true;  
28: else if cohesiveness($p_s, p_t$) == 0 then  
29: cohesiveness($p_s$) = 0;  
30: else  
31: $p_{new} = p_s \cup p_t$;  
32: depth($p_{new}$) = depth($p_s$) + depth($p_t$) + cohesiveness($p_s, p_t$);  
33: for each partition $p_i \in P$ satisfying $i \neq s, t$, new do  
34: cohesiveness($p_{new}, p_i$) = cohesiveness($p_s, p_i$) + cohesiveness($p_t, p_i$);  
35: cohesiveness($p_i, p_{new}$) = cohesiveness($p_{new}, p_i$);  
36: end for  
37: end if  
38: $P = P \cup \{p_{new}\} \setminus \{p_s, p_t\}$;  
39: end while  
40: return $P$;
7.4.4 Optimizing Ontology Alignment with a Branch and Bound Approach

To improve scalability of alignment, we developed an algorithm based on a branch and bound approach. This algorithm considers each concept in ontology $o_i$ and semantically compares it with concepts in ontology $o_j$ seeking to eliminate subtrees in ontology $o_j$ that are not likely to align. The algorithm moves depth first through the graph of $o_j$; if a concept in $o_j$ is not semantically close to the concept under consideration in $o_i$, it is pruned from further consideration along with all of its children. Once the pairs of concepts to align are selected, alignment begins. This algorithm is designed to reduce the time complexity such that $O(n^2)$ is a maximum; the actual complexity is typically much less, depending on the number of semantically close concepts cross-ontology. This Branch and Bound approach is detailed in Algorithm 2.

The most important component of this algorithm is the method by which we evaluate whether two concepts are “semantically close”. Semantic closeness is determined by the relationship that we seek to acquire in the alignment. For example, if we seek to align using subclass relationships, then “semantic closeness” might be defined to be the existence of hyponymy relations among words in class names cross-ontology. If we seek to align using superclass relationships, then we might define “semantic closeness” to be the existence of hypernymy relations among words in class names cross-ontology. Our goal was to select a feature set that would prune unlikely candidates, resulting in a computationally inexpensive yet still highly accurate
Algorithm 2 branchAndBoundAlign($o_i, o_j$)

1: for each root concept $c_j \in o_j$ do
   push $c_j$ onto stack;
2: end for
3: for each concept $c_i \in o_i$ do
4:   while not done do
5:     if stack is empty then
6:       done = true;
7:     else
8:       $c_j = \text{pop stack}$;
9:       if semanticallyClose($c_i, c_j$) then
10:      pairsToCompare.add($c_i, c_j$)
11:      push direct subclasses of $c_j$ onto stack;
12:     end if
13:   end if
14: end while
15: end for
16: Align(pairsToCompare)

approach. Depending on the feature selected, we could expect to encounter varying execution times as well as varying accuracy results. This is discussed more in the next section. As a starting point, we selected one feature, which determines whether two class names contain words that are hyponyms (feature 44, detailed in Appendix C).

The evaluation was performed by aligning the mp and gobp ontologies using subclass relations, with feature 44 used as the semantic distance check. As shown in Figure 7.4, the algorithm results in a dramatic reduction in alignment execution time over the optimized alignment algorithm. The Branch and Bound algorithm aligns in less than an average of 3 minutes as compared to an average of 96 hours with the optimized alignment algorithm. However, there is a cost in execution time in the selection of pairs; the total average time to select pairs to align was approximately 38 hours. This is still a significant reduction in overall execution time, as shown in Figure 7.5.
Figure 7.4: Branch & Bound and Alignment Performance Comparison for up to 1000 Pairs.

To complete our assessment of the value of the Branch and Bound algorithm, we evaluated whether the algorithm results in a reduction in accuracy. To do that, we compared the performance of the Branch and Bound algorithm with the performance of the Optimized Alignment method when aligning the mp and gوب ontologies. To measure accuracy, we used the approximation metrics described in Section 7.3. The evaluation was performed with the SVM models built for the Biomedical domain in an earlier part of this research, described in Chapter 6. We averaged the performance over 3 SVM models. The semantic distance test was performed using feature 44. A comparison of the average match coverage metrics, which estimate recall, are shown in Figure 7.6. These results show that estimated recall is significantly lower, particularly for the second ontology. This is not entirely unexpected, given the nature of the Branch and Bound algorithm; that is, we might expect that less classes in ontology 2 are matched, since they are being pruned in the process. Recall from Section 7.3
that we seek match coverage to be above 60%; clearly, the match coverage of the Branch and Bound algorithm needs improvement. Match ratios, on the other hand, are satisfactory, as shown in Figure 7.7. This suggests that precision is not negatively impacted by the Branch and Bound approach. Recall from Section 7.3 that we seek match ratios in the range of 2-9, with 1 being considered a perfect score. Therefore, these results are quite encouraging. Given the nature of the Branch and Bound algorithm, we would expect that precision should remain as high as the optimized alignment, since the alignment process remains constant. These estimated metrics were validated with random spot checks of the alignments by biological domain experts. Their results were favorable with roughly 80% of the alignments being correct.

The key to improving recall of the Branch and Bound algorithm is to select the optimal feature for use in pruning; that is, the feature that is the best at determining
Figure 7.6: Comparison of large-scale alignment accuracy of Branch & Bound and Optimized Alignment using match coverage, an estimate of recall.

“semantic closeness”. This feature or set of features should result in higher recall but not significantly increase processing time. Therefore we performed experiments to identify optimal semantic distance features for use in the Branch and Bound algorithm for subclass alignment. We compared the following features to determine if they would improve recall.

- Hyponyms between words in class names cross-ontology (used in original results above)
- Hypernyms between words in class names cross-ontology
- Synonyms between words in class names cross-ontology
- Stoilos string metric
- Hyponyms and hypernyms between words in class names cross-ontology
The results are shown in Figure 7.8. Again, these experiments involved aligning the mp and gobp ontologies. We found that using the feature to detect hyponyms between class terms worked the best of all the single features. However, this was the feature originally used, therefore recall was not improved. Combining hypernyms and hyponyms worked best overall, delivering a slight improvement in recall. However, even with the combination of features, recall remains quite low. We believe that utilizing domain resources is the likely solution to improving recall for the Branch and Bound algorithm. The disadvantage, of course, is that the solution will not work in all domains.

In the case of the Stoilos String metric, the Branch and Bound algorithm execution time was significantly higher, more than 50% higher, therefore, we abandoned that approach for semantic closeness. In addition, we didn’t expect that a string
metric would yield good results for semantic distance. In all other cases, execution time was not significant increased over the results discussed earlier in this Section; in fact, in all cases, execution time remained within 5% of the original optimization.

7.5 Experimental Results: Alignment of Large Scale Biomedical Ontologies

In this section, we present the results for aligning large-scale ontologies from the Open Biomedical Ontologies collection using subclass relations. We aligned 8 pairs of ontologies using the SVM models developed earlier, described in Chapter 6. Recall that we use match coverage and match ratios to estimate recall and precision, respectively. A comparison of match coverage can be seen in Figure 7.9. In
Figure 7.9: Match coverage for alignment of large-scale biomedical ontologies, using subclass relationships. Average over 3 SVM models.

In most cases, the alignment performs well. In particular, the NCI Thesaurus (nci) and Mouse Anatomy (ma) alignments result in strong match coverage, as do alignments of the Mammalian Phenotype (mp) and Mouse Anatomy (ma). This is especially encouraging since the SVM models were trained on mp,gobp examples, primarily. Match coverage performance on mp,gobp is also very good.

Match coverage for pairs mp, gomf and mp, gocc is not as strong. An analysis of the results revealed that in fact, many features were extracted for class pairs. For example, in the mp,gomf pair, features were extracted in well over 8000 class pairs. However, in most cases, only single features were identified. The SVM was trained with positive examples that typically possessed multiple features. Therefore, this suggests that the SVM should be trained with a more diverse set of data. In addition, further analysis is needed to identify features that will result in better match coverage for a broader set of ontologies in the Biomedical domain.

Match ratio comparisons of large-scale alignment using subclass relations are
shown in Figure 7.10. Again, in most cases, estimated precision looks very good. This is especially promising given that the SVM models were trained on mp,gobp examples, primarily. The estimated precision in the pair mp, gomf is troublesome; the results show that an inordinate number of matches occur in mp, an unexpected result. An analysis of the results showed that in fact, too many single classes were being matched to multiple classes, an indicator of low precision. Random spot checks confirmed low precision. This suggests that additional features should be identified and incorporated into the SVM to improve precision in a broader set of class pairs.

Note that we supplemented these results with spot checks of the alignments by biological domain experts. Data to spot check was selected randomly over the large sets of pairs. The results were favorable in general, with roughly 80% of the alignments being successful. This is an encouraging result and strengthens our confidence in the use of the estimation metrics.

The complete set of results for aligning with subclass relations is provided in Table 7.3.
Figure 7.10: Match ratios for alignment of large-scale biomedical ontologies, using subclass relationships. Average over 3 SVM models.
Table 7.3: Alignment Evaluation Results for Large Scale Biomedical Ontologies.

<table>
<thead>
<tr>
<th>Ontology Pair</th>
<th>SVM Model</th>
<th>Matched c1</th>
<th>Matched c2</th>
<th>Match Coverage 1</th>
<th>Match Coverage 2</th>
<th>Match Ratio 1</th>
<th>Match Ratio 2</th>
<th>Combined Ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>mp, gobp</td>
<td>1</td>
<td>148</td>
<td>356</td>
<td>0.15</td>
<td>0.36</td>
<td>7.44</td>
<td>3.09</td>
<td>4.37</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>39</td>
<td>72</td>
<td>0.04</td>
<td>0.07</td>
<td>3.23</td>
<td>1.75</td>
<td>2.27</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>151</td>
<td>364</td>
<td>0.15</td>
<td>0.36</td>
<td>7.40</td>
<td>3.07</td>
<td>4.34</td>
</tr>
<tr>
<td>mp, gomf</td>
<td>1</td>
<td>10</td>
<td>97</td>
<td>0.01</td>
<td>0.10</td>
<td>10.80</td>
<td>1.11</td>
<td>2.02</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>11</td>
<td>99</td>
<td>0.01</td>
<td>0.10</td>
<td>10.00</td>
<td>1.11</td>
<td>2.00</td>
</tr>
<tr>
<td>mp, gocc</td>
<td>1</td>
<td>30</td>
<td>44</td>
<td>0.03</td>
<td>0.04</td>
<td>2.43</td>
<td>1.66</td>
<td>1.97</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>7</td>
<td>6</td>
<td>0.01</td>
<td>0.01</td>
<td>1.29</td>
<td>1.50</td>
<td>1.38</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>30</td>
<td>44</td>
<td>0.03</td>
<td>0.04</td>
<td>2.43</td>
<td>1.66</td>
<td>1.97</td>
</tr>
<tr>
<td>mp, ma</td>
<td>1</td>
<td>227</td>
<td>351</td>
<td>0.23</td>
<td>0.35</td>
<td>3.85</td>
<td>2.49</td>
<td>3.02</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>126</td>
<td>141</td>
<td>0.13</td>
<td>0.14</td>
<td>2.13</td>
<td>1.90</td>
<td>2.00</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>232</td>
<td>363</td>
<td>0.23</td>
<td>0.36</td>
<td>3.87</td>
<td>2.47</td>
<td>3.02</td>
</tr>
<tr>
<td>ma, mp</td>
<td>1</td>
<td>275</td>
<td>161</td>
<td>0.28</td>
<td>0.16</td>
<td>2.70</td>
<td>4.61</td>
<td>3.40</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>47</td>
<td>49</td>
<td>0.05</td>
<td>0.05</td>
<td>2.26</td>
<td>2.16</td>
<td>2.21</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>289</td>
<td>166</td>
<td>0.29</td>
<td>0.17</td>
<td>2.65</td>
<td>4.61</td>
<td>3.37</td>
</tr>
<tr>
<td>nci, ma</td>
<td>1</td>
<td>466</td>
<td>579</td>
<td>0.47</td>
<td>0.58</td>
<td>6.27</td>
<td>5.05</td>
<td>5.60</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>279</td>
<td>270</td>
<td>0.28</td>
<td>0.27</td>
<td>1.54</td>
<td>1.59</td>
<td>1.56</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>534</td>
<td>662</td>
<td>0.53</td>
<td>0.66</td>
<td>6.05</td>
<td>4.88</td>
<td>5.41</td>
</tr>
<tr>
<td>ma, nci</td>
<td>1</td>
<td>598</td>
<td>446</td>
<td>0.60</td>
<td>0.45</td>
<td>5.55</td>
<td>7.44</td>
<td>6.36</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>288</td>
<td>264</td>
<td>0.29</td>
<td>0.26</td>
<td>1.65</td>
<td>1.80</td>
<td>1.72</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>677</td>
<td>517</td>
<td>0.68</td>
<td>0.52</td>
<td>5.34</td>
<td>6.97</td>
<td>6.06</td>
</tr>
<tr>
<td>ma, gobp</td>
<td>1</td>
<td>60</td>
<td>45</td>
<td>0.06</td>
<td>0.05</td>
<td>2.53</td>
<td>3.38</td>
<td>2.90</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>22</td>
<td>15</td>
<td>0.02</td>
<td>0.02</td>
<td>2.41</td>
<td>3.53</td>
<td>2.86</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>72</td>
<td>47</td>
<td>0.07</td>
<td>0.05</td>
<td>2.31</td>
<td>3.53</td>
<td>2.79</td>
</tr>
</tbody>
</table>
Clearly, developing scalable infrastructures for ontology processing, reasoning and alignment remains a significant challenge. Infrastructure for ontology processing and reasoning have centered around two approaches; processing native OWL files and processing using relational database management systems. Native OWL stores reduce loading time, but database systems offer maturity and robustness, particularly in query optimization capabilities. Database management systems also have built in capability for transaction processing, access control, indexing, etc. Therefore, in the future, we believe that the approaches will narrow to using relational database management systems for scalable infrastructure.

Solutions to scaling ontology alignment are only starting to emerge and in 2009, scaling alignment remains a challenge. In fact, scalability continues to be one of the grand challenges of ontology construction and use [Hepp et al., 2008] and scaling ontology alignment in particular remains a research area [Giunchiglia et al., 2009]. We analyzed some approaches in this chapter; in particular, we looked closely at Hu’s approach to ontology partitioning using a divide and conquer approach [Hu et al., 2008]. Hu’s algorithm does not consider the qualitative nature of links between classes and the execution time of partitioning is not reported. While pushing the complexity of alignment to an a priori process is a good approach (as it takes steps toward near real-time alignment), we believe that Hu’s approach is too expensive computationally.

We discussed our work with scalability in the processing, reasoning and aligning over large scale biomedical ontologies from the the Open Biomedical Ontologies col-
lection. We described our approach to dynamically creating and populating databases for each ontology. We presented an optimized database schema design and showed that loading time for large-scale ontologies can be reduced. We also built an API with dozens of methods to allow access to the database to support alignment, including a large set of methods that simulate the reasoning capability of OWL. This work allowed alignment of large-scale ontologies to proceed.

We presented a partitioning algorithm that builds on previous work in agglomerative clustering. We attempted to improve the algorithm by considering the qualitative nature of the links in the ontological graphs; we developed a set of functions to measure cohesiveness between members in a partition. However, we found that the partitioning algorithm with agglomerative clustering was too expensive computationally; we found that partitioning ran as long or longer than the original alignment, so we abandoned this approach. Instead, we developed an algorithm that is based on a branch and bound approach. In this novel approach, the algorithm considers each concept in the first ontology and compares it, depth first, to concepts in the second ontology. If the concept pairs are not “semantically close”, then the concept in ontology$_2$ is pruned from further consideration as are its children. This approach caps the maximum time complexity at $O(n^2)$, and typically results in much smaller time complexity, depending on the number of “semantically close” pairs identified.

Since reference alignments for large-scale ontology alignments are not available and not feasible to create, we used emerging metrics from [Kirsten et al., 2007] to approximate precision and recall. We found that the Branch and Bound approach did not significantly reduce precision but it did reduce recall. We worked to identify
functions to optimize the tradeoff between execution time and recall. We found that hyponymy relations between class names worked best as a single feature to identify semantic closeness. Identification of hyponymy relations in conjunction with hypernymy relations cross-ontology worked best to improve recall, without significant increase in execution time. Precision remained at fairly high levels in these experiments.

Finally, we present large-scale alignments of biomedical ontologies using subclass relations. In most cases, the algorithms performed well, according to estimated metrics and random spot checks. Alignment performed well even on ontologies that were not used in training the SVMs, such as the nci,ma and ma,nci pairs. In a few cases, we did find that performance was not satisfactory, as in the case of the mp,gomf and mp,gocc pairs. Analysis of the results showed that performance could be improved in two ways: (1) by adding new features to the SVM (2) by training the SVM with a broader set of training examples. In general, however, our results suggest that our approach is feasible for use in real world applications, in terms of time complexity and accuracy.
Chapter 8

Conclusions and Future Research Directions

8.1 Summary and Conclusions

The goal of this research was to advance ontology alignment in a number of key areas. The results presented in this dissertation show that we contribute to the state of the art in three main areas. First, we extend the information that can be derived in alignment by acquiring non equivalence relations across ontologies, including subclass, superclass and inferred relations. The need for alignment using non equivalence relations was recently recognized by the ontology alignment community as an essential capability [Caracciolo et al., 2008], so we believe that this is an important contribution. In addition, to our knowledge, no data sets exist to test alignment using non equivalence relations; the reference alignments that we have developed are one of the first of their kind, spanning two domains, Academic and Biomedical. We plan to share these reference alignments with the ontology alignment community and on the Web.

Second, we contribute new features and methods to support identification of relationships cross-ontology. We present a set of patterns that utilize the upper ontology OpenCyc in conjunction with OWL semantics. Overall, the patterns were very successful in aligning ontologies that consisted of commonsense terms. In addition,
during the evaluation of the patterns, we discovered that OpenCyC and WordNet perform quite differently. We analyzed the differences and therefore, we also contribute a comparison of the performance of external knowledge sources. In particular, we found that patterns that rely on synonymic relations cross-ontology perform optimally in most cases using WordNet as the source of online knowledge. Patterns that rely on hyponymy and hypernymy were improved by using OpenCyC as the sole knowledge source. The strong performance of OpenCyC, particularly its low error rate, suggests that upper ontologies are a very important resource for ontology alignment.

To extend our approach in more complex domains, such as alignment of Biomedical ontologies, we modeled the ontology alignment problem in a Support Vector Machine. We showed that alignment is enhanced by the SVM, significantly in some cases. The features used in the SVM model are based on patterns, which use semantics, ontological structure and external knowledge sources. The SVM also identifies relationships between classes based on string, linguistic, syntactic and semantic characteristics that exist between each pair. The results of the approach are very promising, with very high accuracy in the Academic domain and satisfactory accuracy in most cases in the Biomedical domain. This suggests that Support Vector Machine technology is a valuable tool in ontology alignment. Precision and recall were not as high for inferred relations, suggesting that additional features of class pairs must be identified to improve performance. To improve alignment performance using inferred relations, we developed a set of heuristics that process class and relation names linguistically. We showed that these heuristics resulted in significant enhancement in accuracy, though the heuristics were tailored specifically for OBO
ontologies and therefore are not likely to work in general. In addition, despite the varying performance of OpenCyc and WordNet found earlier in our work, the SVM was able to perform well while utilizing both knowledge sources since features derived from each are discretely identified and weighed by the SVM. We successfully aligned ontological concepts in two domains using SVMs, including the Academic and Biomedical domains.

Finally, we investigated scalability in the processing, reasoning and alignment of large-scale ontologies. We built a scalable infrastructure to process and align large scale ontologies. This infrastructure consists of an optimized database along with logic designed to mimic the most widely used capabilities of OWL as well as other capabilities to support alignment. We showed that this infrastructure improved database loading times significantly.

To address scalability in alignment, we contributed a new algorithm for scaling ontology alignment, in which a Branch and Bound algorithm is tailored to select optimal pairs of classes to align. The algorithm moves depth first through the second ontology, pruning those classes that are not “semantically close” to classes in the first ontology. The heart of the algorithm is the function to measure semantic distance between each class pair. We showed that this algorithm significantly reduces processing time without major reduction in precision. However, the Branch and Bound algorithm did significantly reduce recall. In an attempt to improve recall, we evaluated various features for use in determining semantic distance. We found that the combination of detecting hyponyms and hypernyms cross-ontology did improve recall slightly but more improvement is still needed. We believe that use of domain-specific
knowledge sources are the next logical step in improving recall in the Branch and Bound approach. We also aligned several large-scale ontologies in the Biomedical domain using subclass relations and evaluated those results. We found that accuracy was strong in most areas, in fact, strongest in class pairs that did not comprise training data. This is a very encouraging result. However, some of the results suggested that the SVM should be trained with a more diverse set of training examples and that new features in the SVM could improve accuracy in a broader set of ontology alignments.

There are many challenges that remain in ontology alignment. In the next section, we describe the difficulty in automated ontology alignment, particularly those areas that are likely to remain a challenge for some time.

8.2 Continuing Challenges in Ontology Alignment

8.2.1 Difficulty of the Alignment Problem

In order to successfully discover relationships between ontological entities, the semantics of the entities must be understood. Automated extraction of semantics from entity names is a difficult problem for many reasons. First, the expression of semantics is traditionally implicit (e.g. contained in the structure, contained in the software that uses the data, contained in the modelers’ minds, etc.) and design styles of modelers vary greatly. Often, models are designed with a particular end use in mind; so, even within a single domain, models can vary greatly depending on how they are to be used. Extraction of intended usage is typically not fully possible
without direct intervention from human experts. In addition, there are some obstacles that cannot be completely solved in automated ontology alignment. Poor naming conventions and poor database design will impede automated solutions to alignment. For example, when a database field (row and/or column) is used for multiple, varying purposes in practice\(^1\), there may not be a way to extract meaningful semantics. It is interesting to note that these challenges are often not easily solved by humans, not to mention automated methods. For all of these reasons, development of a broadly applicable solution to automated ontology alignment remains a research challenge [Euzenat and Shvaiko, 2007].

The subtlety of semantics and usage of terms will remain a challenge, as described in Chapter 5. Best practices in the development of ontologies and usage of standards, where applicable, could be a great approach to mitigating the challenge. For example, ontology authors could keep in mind a broader use of the knowledge, including use by machines. With that in mind, an ontology class could be named *ChairPerson* instead of *Chair*. Additionally, meta data could play a role in describing the context of the knowledge in an ontology. Though ontologies, by nature, are intended to be self-describing, often in practice, are not. Therefore, the addition of meta data to ontologies could bring us closer to more effective methods of achieving the goal of

\(^1\)Unfortunately, in our experience, this practice is fairly common and is used to avoid software changes to systems in operations. For example, in industry, to remain competitive, marketing organizations often demand that no software changes be required in order to more rapidly deliver new capabilities to customers. This forced reuse of existing database fields for new purposes skews the semantics and usage of the field.
self-describing ontologies. One such approach is presented in Oro and Ruffolo [2008].

8.2.2 Evaluation Issues

Methods for evaluating ontologies, particularly large-scale ontologies, are likely to continue to evolve for some time. There are two key challenges in evaluating ontology alignment techniques. First, alignment tasks can be very different; they depend on the domain, specific data, author, and other factors. This can result in great variance in accuracy across techniques [Euzenat and Shvaiko, 2007]. Some alignment systems may perform very well over some data sources and not on others. Therefore, to perform effective evaluations, standard data sets must be used. Second, some traditional metrics may not be appropriate for ontology alignment [Euzenat and Shvaiko, 2007]. We discuss this further in Section 8.2.3.

In recent years, there have been a number of initiatives to establish standard mechanisms and data sets to evaluate ontology alignment techniques. The premier forum for evaluating ontology alignment has been the Ontology Alignment Evaluation Initiative (OAEI), held each year since 2005. Other organized evaluation forums include the Information Interpretation and Integration Conference (I3CON)\textsuperscript{2} held in 2004, and the Evaluation of Ontology Based Tools (EON) also in 2004\textsuperscript{3} [Sure et al., 2004].

We describe here the latest evaluation techniques used in OAEI 2008 and OAEI 2007. In this evaluations, a set of test tracks is defined to be used to evaluate align-

\textsuperscript{2}http://www.atl.external.lmco.com/projects/ontology/i3con.html
\textsuperscript{3}http://oaei.ontologymatching.org/2004/Contest/
ment techniques. There are as follows.

1. Comparison (or benchmark)

2. Expressive ontologies

3. Directories and thesauri

4. Consensus workshop

The Comparison series of tests is designed to identify areas of strengths and weaknesses in each approach. This is accomplished by a series of tests in the bibliography domain that are increasingly difficult. The first set of benchmark tests, known as the ”1xx” series, verifies basic capabilities. First, quality is checked by aligning the bibliography ontology to itself, which should result in 100% alignment. Next, the bibliography ontology is to be aligned to a completely irrelevant ontology, specifically, a food ontology. This should result in no alignments identified. Further tests in the 1xx series test alignment by varying language generalization and restriction. The 2xx series consists of tests that remove or drastically modify the labels of ontological component and the structure. Foreign names are used, and hierarchies are missing, flattened or expanded. Other tests involve the removal of instances, introduce differences in units, remove properties, etc. These tests are useful to pinpoint the strengths and weaknesses of each approach. Finally, the 3xx series involves comparisons with ontologies in use in the bibliographic domain.

The Expressive ontologies track involves alignment tests with real-world ontologies in the area of human and animal anatomy.
The Directories and thesauri test set involves aligning the directories of Web sites. Also there are tests to align ontologies in the food, environment and library domains. Expected results are developed by domain experts in these tests. Finally, the consensus workshop tests involve alignment of conference organization ontologies. Despite these advances, evaluation of alignment systems remains a future research challenge [Euzenat and Shvaiko, 2007], [Ehrig, 2007]. In particular, there remains a need for high quality data sets that capture different types of alignment challenges, definition of application specific settings to support comparisons, and evaluation measures for different types of tasks [Euzenat and Shvaiko, 2007].

Note that we were unable to test our application using these test sets since they are designed to evaluate strictly equivalence relations.

8.2.3 Metrics Issues

There are also significant issues with methods to measure the accuracy of ontology alignment. Precision and recall are widely used to measure the accuracy of classification systems. However, these measures take an ”all or nothing” approach to performance and do not consider degrees of accuracy. For example, two ontology alignment techniques could fail to match components, but one could be much higher in score than another, yet both would be rated as having similar accuracy. This notion of ”near misses” needs to be captured. Therefore, use of precision and recall may not be appropriate for comparing the accuracy of ontology alignment approaches [Euzenat and Shvaiko, 2007]. This suggests that in our work, the precision and recall
reported represents a minimum measure of performance.

To address this problem, new methods for calculating precision and recall are being proposed to distinguish finer degrees of accuracy. For example, [Euzenat and Shvaiko, 2007] propose a notion of relaxed precision and recall. In this method, the proximity of alignments are measured by considering the raw scores of confidences.

In addition, use of precision and recall requires a reference alignment. As we encountered in our work to align large-scale ontologies, manual construction of reference alignments is very laborious and extremely difficult. This is particularly the case in specialized domains such as the Biomedical domain. Increasingly, alignment approaches are using estimates of performance, including rough approximations for recall and precision and relative quality estimates. Examples of this are the metrics we used from [Kirsten et al., 2007], match coverage and match ratios, to estimate precision and recall. Other methods are proposed in [Caracciolo et al., 2008] and [Euzenat et al., 2007]. In late 2008, the OAEI reported that their future plans include “improving the measures to go beyond precision and recall” [Caracciolo et al., 2008]. Therefore, evaluation metrics for ontology alignment is still very much a research area.

In addition, some domains offer significant challenges in alignment since ontological entities can be quite complex. For example, in the Biomedical domain, identification of non-trivial correspondences sometimes requires medical background knowledge [Caracciolo et al., 2008]; to distinguish best performing alignment approaches in these cases, the OAEI applies an additional measure known as recall+. Recall+ measures how many non-trivial correspondences are found in an alignment
[Caracciolo et al., 2008].

8.2.4 Scalability Issues

As we saw in Chapter 7, there are major challenges in scaling ontology processing, reasoning and alignment. Two leading approaches to scaling ontology processing and reasoning are (1) use of native OWL stores and (2) use of databases. We believe that databases along with methods to simulate ontological reasoning, will emerge as the preferred approach in the future. Solutions to scaling ontology alignment are only starting to emerge in the past 2 years. As we saw in Chapter 7, many of these approaches are domain-specific or they push the complexity of alignment to an a priori process and claim significant reductions in execution time. We proposed an algorithm that successfully reduced execution time but at the cost of recall. We tested a broad set of features to improve recall but had minimal success. We suspect that use of domain resources could improve recall without increase to execution time, but then our goal of a domain-independent solution is not met. We submit that our analysis of the work of others, as well as our own results demonstrate that scalability in ontology alignment is one of the primary areas for original contribution in the future.

We can envision a future Web that requires software entities that meet for the first time to interact in a semantically meaningful way. Ontology alignment could play an important role in this type of dynamic mediation over the Web. If ontology alignment is to contribute to this future Web, then alignment services must perform in real-time. Therefore, scalability of alignment is an essential part of the way forward.
In the next Section, we describe ideas for how to further this research.

8.3 Future Research Directions

There are a significant number of ways to enhance and continue this research. In this Chapter, we describe some of the key areas we believe should be pursued. First, we describe the use of ontology alignment in the Biomedical domain, focusing on three specific applications. Next, we describe alignment for use in rapid enterprise integration; in particular, we discuss how alignment might be applied to integrate and harness the collective knowledge across multiple data sources in the Sports Medicine domain. In addition, we discuss the role of alignment in dynamic Web service composition and mediation; and, we present an idea for role-based alignment in which knowledge about the user might be leveraged to enhance the alignment process. Finally, we conclude with a broad set of additional areas to further advance this research.

8.3.1 Alignment in the Biomedical Domain

There are three primary areas in the Biomedical domain that could benefit from ontology alignment techniques. First, the process of data analysis could be enhanced for biological researchers. Second, a hypothesis generation tool could be developed, based on alignments of data that represent experimental results in gene modifications. Third, we suggest that semantic-based ontology alignment could be applied to enhance gene clustering techniques. We discuss each of these ideas in detail in the next three sections.
8.3.1.1 Enhancing Data Analysis for Biological Researchers

In the past decade, high-throughput screening methods have revolutionized the analysis of biological systems. Fast and reliable methods for simultaneous analysis of all biological molecules are now available. However, they are limited by the lack of automated methodologies for analyzing the data generated from high-throughput experimentation. A variety of Web-based tools have been developed to compile the wealth of information that has emerged in the biomedical domain. An example of the tools that are widely used is the Mouse Genome Informatics (MGI) site\(^4\). While these tools are useful for single gene and protein queries, they are not ideal for analyzing lists of multiple genes and proteins. Furthermore, although there are efforts to make the various Web-based knowledge domains interoperable, the integration of this information is still in its infancy. For example, a biological researcher may want to use the MGI site to query for similarities among a list of genes that were identified from high-throughput experimentation, such as transcriptional profiling and proteomics. Transcriptional profiling, also known as gene expression profiling, is the measurement of the activity of thousands of genes at once in order to create a global picture of cellular function\(^5\). The term proteome defines the entire protein complement in a cell, comprising protein activities, modifications and interactions. Encoded proteins carry out most biological functions, so proteomics is essential to understanding how cells work\(^6\). Therefore, both of these functions are essential to the biological researcher.

\(^4\)http://www.informatics.jax.org/javawi2/servlet/WIFetch?page=batchQF
While the MGI site provides a means for conducting batch queries, it can only search one ontology at a time, such as the Gene Ontology or Mammalian Phenotype Ontology. Furthermore, it typically returns a copious amount of data when querying from each ontology. There is no way now to automatically relate the data retrieved in any way. This work must be performed manually. The current search page for MGI is shown in Figure 8.1.

As an illustration, we queried the Mammalian Phenotype (MP) ontology for three genes, *p53*, *p16* and *pten*; the results of this query are shown in Figure 8.2. The query for three genes in just a single ontology returns 467 matching rows, involving pages of data. The researcher must search each ontology separately, perusing the hundreds, perhaps thousands of resultant matches to determine the similarities and differences between the returned results.
A potential solution for this problem could make use of aligned concepts within biomedical ontologies. A notional solution using alignment is shown in Figure 8.3. In this application, biomedical ontologies are aligned a priori using equivalence, subclass and superclass relationships, as well as using relationships that represent differences between the data. The search capability provides information from the ontologies of interest; the ontology alignment is used to identify common characteristics as well as differences between each ontology. The analyst could then narrow in more quickly on the areas of interest.

In addition to the above ideas, we believe that the power of OpenCyc could be leveraged in the Biomedical domain. We saw in Chapters 5 and 6 that OpenCyc delivered superior performance as an external knowledge source for commonsense
Applying ontology alignment to enhance searches of biomedical ontologies. To take advantage of that power, it might prove useful to convert biomedical ontologies into a Cyc-like knowledge base. The knowledge base could then leverage the power of Cyc in querying and reasoning over the information. This could be a “BioCyc” of sorts and could transform the tasks involved in biological data analysis. This type of reasoning capability could also be integrated with the existing search capabilities of the OBO ontologies.

### 8.3.1.2 Hypothesis Generation for Human Disease Research

Another compelling example that illustrates the potential power of this technology is the use of ontology alignment techniques to integrate knowledge from biomedical experiments, particularly across species. For example, the Knockout Mouse
Project (KOMP)\textsuperscript{7} is an initiative of the National Institutes of Health (NIH) to generate a resource comprised of mouse embryonic stem cells containing mutations in every gene in the mouse genome. The resulting data generated from experimentation conducted with this collection will comprise a catalog of mutants and the observed mouse phenotypes, which will ultimately enhance the elucidation of human disease development, according to the NIH site for KOMP\textsuperscript{8}. We submit that ontology alignment techniques could enrich the results of the KOMP by aligning gene and phenotype information across multiple existing ontologies, as shown in Figure 8.4. The collective knowledge obtained through such alignment could be used to generate hypotheses; for example, if gene $x$ is modified in a knockout mouse, resulting in phenotype $y$, then perhaps gene $x$ in humans causes phenotype $y$. Note that the KOMP initiative has only recently started, so the information is not yet available. Once complete, the MGI database\textsuperscript{9} will host the resultant knockout mouse data.

\subsection*{8.3.1.3 Complementing Gene Expression Clustering Techniques}

There has been success in identifying groups of genes that manifest similar and dissimilar expression patterns [Das et al., 2009], [Das et al., 2008] and [Das et al., 2007]. For example, in [Das et al., 2009], a parameter-less clustering technique is proposed that uses a dynamically calculated threshold to assign cluster membership.

\begin{thebibliography}{9}
\bibitem{KOMP} http://www.knockoutmouse.org/
\bibitem{NIH} http://www.nih.gov/science/models/mouse/knockout/komp.html
\bibitem{MGI} http://www.informatics.jax.org
\end{thebibliography}
Figure 8.4: Notional example of ontology alignment in the biomedical domain. Knockout Mouse Project results could be enriched through aligned gene and phenotype knowledge sources.

Also, a dissimilarity measure is proposed that is less susceptible to outliers. These methods are very encouraging for gene expression time series clustering, however, the methods result in somewhat coarse clusters. There is a need to further process the clusters qualitatively to identify key similarities and dissimilarities in functions. We submit that ontology alignment could complement the process by performing a secondary semantic-based check of relations between gene expression data.

### 8.3.2 Alignment for Rapid Enterprise Integration

Ontology alignment can also be applied for integration of enterprise data. There is an increasing need across government and industry for rapid integration of enterprise data. As businesses merge, there is a need to integrate customer, accounting and other type of data. As government agencies seek to cooperate more than ever before, there is a need to share information cross-organizationally in a meaningful
way. In this section, we highlight a real world application for ontology alignment that has been proposed as a National Science Foundation (NSF) research project by UCCS [Lewis et al., 2008]. The goal of the research is to apply ontology alignment to integrate Sports Medicine databases to support the U.S. Olympic Training Centers.

Performance in sports is determined by many factors. These factors include physiological, biomechanical, motor skill, psychological and psycho-social, nutritional, and equipment-related factors [Astrand et al., 2003]. Sports science teams support coaches and athletes by collecting information on routine physicals, screening, and monitoring of performance and health indicators. However, the data collected from different US Olympic Training Centers are stored using widely varying syntax and semantics. There are only limited standards for data collection across sports science organizations. Therefore it is difficult for Sport Science to take advantage of the collective knowledge spread across heterogeneous formats and platforms. The approach to care and performance enhancement of athletes is not optimal. In fact, recent data that examine trends of world record improvements have partly been explained by the integration and advancement of sport science as an invaluable part of athlete preparation [Lippi et al., 2008]. Therefore, it seems apparent that integrating heterogeneous data on athlete performance in an infrastructure that connects all US Olympic Training Centers will improve sport science and medicine research and would allow even a greater success of US Olympic and elite athletes [Lewis et al., 2008]. We submit that ontology alignment can play a major role in achieving this vision.
Another application for ontology alignment is the use of aligned concepts to support dynamic Web service composition and mediation. Only a few proposed implementations of dynamic Web service composition and mediation have been put forth. [Hibner and Zielinski, 2007] present a semantic-based approach to automated Web service composition and adaptation. They employ a backward-chaining reasoning method to dynamically compose and adapt services using WSMO and ESB technology. [Castano et al., 2003] propose ontology-based mediation in which wrappers are employed to provide an integrated view over multiple heterogeneous sources. Marconi et al. [2007] describe the application of automated service composition developed to support Amazon, though this approach does not apply semantics. There has been some work on developing ontology alignment to support agent communication and negotiation [van Eijk et al., 2001], [Bailin and Truszkowski, 2002], [Wiesman et al., 2002], [Wang and Gasser, 2002], [Enzenat, 2005]. Clearly, the area of dynamic Web service composition and mediation is a continuing research challenge.

One approach to service composition might build on [Hibner and Zielinski, 2007] but make use of alignment of input and output ontologies. In Hibner and Zielinski, a goal-driven approach is employed that uses backward-chaining reasoning. Essentially, they match Web service post-conditions to the conditions required to meet the current goal. This is iterative since a composition may involve chaining more than one Web service. We would like to explore extending this process, to include matching on the
concepts present in each condition; i.e., using ontology alignment results to compose service chains by matching the output of one service to the input of another. One possible approach to a composition service is shown at a high level in Figure 8.5. In this example, the input of domain\textsubscript{2} is matched to the output of domain\textsubscript{1}, suggesting a possible service chain. There are many other considerations when building such chains. For example, syntactic format and units must also match. However, ontology alignment contributes to the most difficult aspect of the problem - semantic matching.

![Diagram of service composition](image)

**Figure 8.5:** Application of alignment: Web Service Composition.

In addition, the result of the alignment process could be incorporated into an alignment Web service that supports mediation between services, as shown in Figure 8.6. In this case, the alignment serves to mediate between concepts that are exposed by Web services across multiple domains. We envision that such a media-
tion service could be supported by a set of methods that could operate cross-domain.

These methods are provided in Table 8.1 along with just a small set of examples of use cases.

Figure 8.6: Application of alignment: Web Service Mediation.
<table>
<thead>
<tr>
<th>Service</th>
<th>Function</th>
<th>Example use cases</th>
</tr>
</thead>
<tbody>
<tr>
<td>getAllAlignments(x)</td>
<td>Given a term x, provide terms that align</td>
<td>Web Service Registries use this to identify equivalent terms, broaden the search for relevant Web services. Service mediation uses this to obtain related terms across domains.</td>
</tr>
<tr>
<td>getAllSynonyms(x)</td>
<td>Given a term x, what are its synonyms</td>
<td>Web service mediation uses this to support queries cross-domain. Web service composition uses this to construct service chains.</td>
</tr>
<tr>
<td>getAllSubclasses(x)</td>
<td>Given a term x, what are its hyponyms</td>
<td>Used by biologists to identify hierarchies among genes and proteins. Used in Sports Science to understand relations between athlete performance and contributing factors.</td>
</tr>
<tr>
<td>getAllSuperclasses(x)</td>
<td>Given a term x, what are its hypernyms</td>
<td>Used by biologists to identify hierarchies among genes and proteins. Used in Sports Science to understand relations between athlete performance and contributing factors.</td>
</tr>
<tr>
<td>getAllEquivalentTerms(x)</td>
<td>Given a term x, what are its equivalent terms</td>
<td>Web service mediation uses this to support queries cross-domain. Web service composition uses this to construct service chains. Used by biologists to understand similarities between genes and proteins.</td>
</tr>
<tr>
<td>getAllDisjointTerms(x)</td>
<td>Given a term x, what are its disjoint terms</td>
<td>Used by biologists to identify differences between genes and proteins. Used in Sports Science to understand differences between athlete performance and contributing factors.</td>
</tr>
<tr>
<td>getAllTriples(x, y)</td>
<td>Given terms x and y, return relationships between the two</td>
<td>Used by biologists to understand relations between genes and proteins. Used in Sports Science to understand relations between factors that drive athlete performance.</td>
</tr>
<tr>
<td>align(o1, o2)</td>
<td>Given 2 ontologies, what are the aligned terms</td>
<td>Used by analyst to identify relations across ontologies. Used by biologists to generate hypotheses for disease research.</td>
</tr>
</tbody>
</table>
8.3.4 Role-based Ontology Alignment

As described earlier in this Chapter, one of the key challenges in automated alignment is the extraction of semantics and intended usage of concepts. In addition, there may be many desired usages of alignment which could result in different types of requirements for alignment. One idea that seems ripe for research is the idea of role-based ontology alignment. That is, could information describing the user help set context for the intended usage of the alignment and therefore enrich the alignment process? To our knowledge, these types of questions have not yet been explored. This notion of role-based alignment is shown in Figure 8.7. We envision that data used to infer roles could be based on explicit information stated by the user as well as meta data automatically obtained by the system.

Figure 8.7: Role based alignment could utilize user meta data to enhance the alignment process.
8.3.5 Additional Directions

In addition to the compelling areas described above, there are other directions for this research. For example, as discussed in Chapter 7, scalability remains a major challenge in ontology alignment. Pursuing new algorithms to scale alignment without reducing overall accuracy is an area of research that is ripe for contributions. The Branch and Bound algorithm described in Chapter 7 is a promising lead in the search for a scalable approach to alignment. Therefore, improving upon this algorithm would be an excellent area to explore. In particular, could we strengthen the semantic similarity check to better pick class pairs to align? Could we develop a scalable approach that runs a Branch and Bound algorithm and improves runtime efficiency further with dynamic multithreading? It might also be interesting to test and compare industry-developed as well as open source tools to compare performance.

In addition, we would like to see this work extended to align additional ontological entities; specifically, aligning relations (properties) as well as classes. We think that alignment accuracy may be improved by using the oboInOwl:Synonym feature in OBO ontologies described in Chapter 4. In addition, using natural language techniques to leverage the information in oboInOwl:Definition may also enhance the performance of alignment, particularly in complex domains such as the Biomedical domain. Wikipedia and links between its pages may also prove a useful source of external knowledge, as suggested in [Stoutenburg and Kalita, 2009].

There are many additional relations that could be used to align ontologies, such as disjoint, partOf, hasPart. As discussed earlier in this chapter, learning disjoint
relations could aid in biological data analysis. It would also be very useful to validate
the findings in this work in new domains, beyond biomedicine. Use of ResearchCyc,
which contains significantly more knowledge than OpenCyc, would likely enhance
alignment performance. It would be interesting to apply unsupervised techniques
prior to alignment to select optimal features for alignment.

Finally, it might be interesting to explore optimizing this approach for use in the
Biomedical domain by using domain knowledge sources. It would useful to compare
the performance of each approach and determine if domain specific knowledge sources
improve accuracy. The performance cost of additional consultation of knowledge
sources would also have to be evaluated.
Appendix A

Reference Alignment Examples in the Academic Domain

In this appendix, we provide examples of the different types of alignments used in the reference alignment of ontologies in the Academic domain.

We start by providing a simple example in Figure A.1. In this example, ontology$_1$ and ontology$_4$ from the Academic domain (as described in Chapter 4) are aligned using subclass relations. For simplicity, only a few classes and relations are shown. In this case, we expect that the subclasses of Person from ontology$_4$ are aligned as subclasses of Person in ontology$_1$. We expect that the subclasses of Event in ontology$_4$ align as subclasses of Event in ontology$_1$. We expect that Talk in ontology$_1$ aligns as a subclass of Event in ontology$_4$. Finally, though Place is not a strict subclass of Location, we expect that some relation should be discovered between Location and Place. Note that the reference alignments for superclass relations are inverses of the subclass relation alignments.

Another example is provided in Figure A.2. In this example, the same two ontologies are used but the reference alignment includes inferred relations. Only a few classes and relations are shown; and yet, the diagram gets quite complex very quickly.
Figure A.1: Example reference alignment in the Academic domain.
Figure A.2: Example reference alignment of inferred relations in the Academic domain.
All reference alignments used in this work were manually generated, though automation was used to speed the process. The diagrams shown above are represented as comma separated data to the alignment software. The format of the semi-automatically generated reference alignments are shown in Figure A.3.
Figure A.3: Data representation of the reference alignment, semi-automatically generated.
Appendix B

Reference Alignment Examples in the Biomedical Domain

In this appendix, we provide examples of alignments used in the reference alignment of ontologies in the Biomedical domain.

The first example in Figure B.1 is a subset of the Platelet Activation test case mentioned in Chapter 6. In this case, we align the Mammalian Phenotype (MP) ontology with the Gene Ontology (GO). We expect that the class in MP *AbnormalPlateletActivation* will align as subclass of *PlateletActivation*, based on the linguistic relationship between the two class names. Since subclass is transitive, we expect that *AbnormalPlateletActivation* will also align as a subclass of *Homostasis*. In addition, *AbnormalPlateletActivation* is expected to align to *BloodCoagulation* using the *partOf* relation, since *PlateletActivation* is *partOf* *BloodCoagulation*. Since *AbnormalPlateletActivation* is a subclass of *PlateletActivation*, we would also expect that those classes that have directed properties to *PlateletActivation* would align with those directed properties to *AbnormalPlateletActivation*. That is, we expect that *RegulationOfPlateletActivation* , *NegativeRegulationOfPlateletActivation* and *PositiveRegulationOfPlateletActivation* align to *AbnormalPlateletActivation* with the relations *regulates*, *negativelyRegulates* and *positivelyRegulates*, respectively.
Figure B.1: Example reference alignment in the Biomedical domain: Platelet Activation.
The second example we present is in the area of Bone Remodeling, again aligning using the MP and GO ontologies. The reference alignment is shown in Figure B.2. In this case, we seek to discover that AbnormalBoneRemodeling is a subclass of BoneRemodeling and that AbnormalBoneResorption is a subclass of BoneResorption. In addition, DecreasedBoneResorption and IncreasedBoneResorption should be identified as subclasses of BoneResorption. We should also infer the relationships regulates, negativelyRegulates and positivelyRegulates cross-ontology in a number of cases, as shown in Figure B.2.
Figure B.2: Example reference alignment in the Biomedical domain: Bone Remodeling.
Appendix C

Evidence Primitives for Ontology Alignment

In this Appendix, we provide the complete set of evidence primitives used in ontology alignment. These evidence primitives combine to form features used in the Support Vector Machine. Evidence primitives and features are described in detail in Chapter 6.
<table>
<thead>
<tr>
<th>Evidence Primitive</th>
<th>Type</th>
<th>Explanation</th>
<th>Subclass</th>
<th>Superclass</th>
<th>Inferred Relations</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 $&lt; c_{ik}, c_{im}, hyponymy, OWL &gt;$</td>
<td>Pattern</td>
<td>First condition in multiple patterns; indicates that class $c_{ik}$ has superclasses in ontology $i$</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>2 $&lt; c_{ik}, c_{im}, p, OWL &gt;$</td>
<td>Pattern</td>
<td>First part of Relations in R Pattern 1, 2, 3, indicates that $p$ exists between class $c_{ik}$ and some other class $c_{im}$</td>
<td></td>
<td></td>
<td>x</td>
</tr>
<tr>
<td>3 $&lt; c_{im}, c_{jl}, synonymy, WordNet &gt;$</td>
<td>Pattern</td>
<td>Second condition in subclass and superclass patterns 1; synonymy exists cross-ontology</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>4 $&lt; c_{im}, c_{jl}, synonymy, OpenCyc &gt;$</td>
<td>Pattern</td>
<td>Second condition in subclass and superclass patterns 1; synonymy exists cross-ontology</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>5 $&lt; c_{im}, c_{jl}, hyponymy, WordNet &gt;$</td>
<td>Pattern</td>
<td>Second condition in subclass and superclass patterns 2; hyponymy exists cross-ontology</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>6 $&lt; c_{im}, c_{jl}, hyponymy, OpenCyc &gt;$</td>
<td>Pattern</td>
<td>Second condition in subclass and superclass patterns 2; hyponymy exists cross-ontology</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
</tbody>
</table>

Continued on next page
<table>
<thead>
<tr>
<th>Evidence Primitive</th>
<th>Type</th>
<th>Explanation</th>
<th>Subclass</th>
<th>Superclass</th>
<th>Inferred Relations</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>Pattern</td>
<td>Second condition in subclass and superclass patterns 3; hypernymy exists cross-ontology</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>8</td>
<td>Pattern</td>
<td>Second condition in subclass and superclass patterns 3; hypernymy exists cross-ontology</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>9</td>
<td>Linguistic</td>
<td>Direct hyponymy relation exists between classes</td>
<td>x</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>Semantic</td>
<td>Direct hyponymy relation exists between classes</td>
<td>x</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>Linguistic</td>
<td>Direct hyponymy relation exists between classes</td>
<td>x</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>Semantic</td>
<td>Direct hyponymy relation exists between classes</td>
<td>x</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>Semantic</td>
<td>Direct relation $p$ exists between classes</td>
<td>x</td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>Semantic</td>
<td>Direct relation $p$ exists between classes</td>
<td>x</td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>String</td>
<td>Scores string distance based on maximum substring; threshold &gt; 0.5</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>16</td>
<td>String</td>
<td>String distance between class and relation names; threshold &gt; 0.5</td>
<td>x</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Evidence</td>
<td>Type</td>
<td>Explanation</td>
<td>Subclass</td>
<td>Superclass</td>
<td>Inferred Relations</td>
</tr>
<tr>
<td>----------</td>
<td>------</td>
<td>-------------</td>
<td>----------</td>
<td>------------</td>
<td>--------------------</td>
</tr>
<tr>
<td>$w_{ik}$, relationwords, stringDistanceMeasure, String</td>
<td>String</td>
<td>String distance between words in class and relation names; threshold &gt; 0.5</td>
<td>x</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$c_{ik}$, $c_{jl}$, endsInSameSubstring, method</td>
<td>String</td>
<td>Classes end in same substring (e.g., BSStudent, Student; threshold of 5)</td>
<td>x</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>$c_{ik}$, $c_{jl}$, beginsWithSameSubstring, method</td>
<td>String</td>
<td>Classes begin with same substring (e.g., Student, StudentOfPhysics; threshold of 5)</td>
<td>x</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>$c_{ik}$, $c_{jl}$, subPrefix, method</td>
<td>String</td>
<td>First class has sub-prefix of second class name (e.g., subcategory, category)</td>
<td>x</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>$c_{ik}$, $c_{jl}$, sameName, method</td>
<td>String</td>
<td>Class names match exactly</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>$c_{ik}$, $c_{jl}$, startsWithSameWords, method</td>
<td>Linguistic</td>
<td>Classes start with same words; threshold of 2</td>
<td>x</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$c_{ik}$, $c_{jl}$, endsInSameWords, method</td>
<td>Linguistic</td>
<td>Class names end in same words; threshold of 2</td>
<td>x</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$c_{ik}$, $c_{jl}$, endsInSameWordsPreceeded ByAdjective, method</td>
<td>String, Linguistic</td>
<td>Class names end in same words and are preceded by an adjective (e.g., AbnormalPlateletActivation, PlateletActivation); threshold of 2</td>
<td>x</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$c_{ik}$, $c_{jl}$, subclassesEndInSameWords PreceededByAdjective, method</td>
<td>Linguistic</td>
<td>Term has subclasses that end in same word, preceeded by adjective</td>
<td>x</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Continued on next page</td>
</tr>
<tr>
<td>Evidence Primitive</td>
<td>Type</td>
<td>Explanation</td>
<td>Subclass</td>
<td>Superclass</td>
<td>Inferred Relations</td>
</tr>
<tr>
<td>--------------------</td>
<td>------</td>
<td>-------------</td>
<td>----------</td>
<td>------------</td>
<td>------------------</td>
</tr>
<tr>
<td>26 (&lt; \ c_{ik}, c_{jl}, \ \text{directSubclassesEndInSame} \ \text{WordsPreceededByAdjective, method} &gt;)</td>
<td>Linguistic</td>
<td>Term has subclasses that end in same word, preceeded by adjective</td>
<td>x</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>27 (&lt; \ c_{ik}, c_{jl}, \ \text{containsSubstringPreceededBy} \ \text{Adjective, method} &gt;)</td>
<td>String, Linguistic</td>
<td>Class contains substring preceeded by adjective (e.g., AbnormalPlateletActivation, CellActivation)</td>
<td>x</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>28 (&lt; \ c_{ik}, c_{jl}, \ \text{startsWithSameWordsPreceededBy} \ \text{Adjective, method} &gt;)</td>
<td>String, Linguistic, Bag</td>
<td>Class contains substring preceeded by adjective (e.g., AbnormalPlateletActivation, CellActivation)</td>
<td>x</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>29 (&lt; \ c_{jl}, c_{jm}, \ \text{hyponymy, OWL} &gt;)</td>
<td>Pattern</td>
<td>First condition in multiple patterns; indicates that class (c_{jl}) has superclasses in ontology</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>30 (&lt; \ c_{ik}, c_{jm}, \ \text{synonymy, WordNet} &gt;)</td>
<td>Linguistic</td>
<td>Second condition in subclass and superclass patterns 1; synonymy exists cross-ontology</td>
<td>x</td>
<td></td>
<td></td>
</tr>
<tr>
<td>31 (&lt; \ c_{ik}, c_{jm}, \ \text{synonymy, OpenCyc} &gt;)</td>
<td>Semantic</td>
<td>Second condition in subclass and superclass patterns 1; synonymy exists cross-ontology</td>
<td>x</td>
<td></td>
<td></td>
</tr>
<tr>
<td>32 (&lt; \ c_{ik}, c_{jm}, \ \text{hyponymy, WordNet} &gt;)</td>
<td>Linguistic</td>
<td>Second condition in subclass and superclass patterns 1; synonymy exists cross-ontology</td>
<td>x</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Continued on next page
<table>
<thead>
<tr>
<th>Evidence Primitive</th>
<th>Type</th>
<th>Explanation</th>
<th>Subclass</th>
<th>Superclass</th>
<th>Inferred Relations</th>
</tr>
</thead>
<tbody>
<tr>
<td>33 (&lt;c_{ik}, c_{jm}, hyponymy, OpenCyc &gt;)</td>
<td>Semantic</td>
<td>Second condition in subclass and super-class patterns 1; synonymy exists cross-ontology</td>
<td></td>
<td></td>
<td>x</td>
</tr>
<tr>
<td>34 (&lt;c_{ik}, c_{jm}, hypernymy, WordNet &gt;)</td>
<td>Linguistic</td>
<td>Second condition in subclass and super-class patterns 1; synonymy exists cross-ontology</td>
<td></td>
<td></td>
<td>x</td>
</tr>
<tr>
<td>35 (&lt;c_{ik}, c_{jm}, hypernymy, OpenCyc &gt;)</td>
<td>Semantic</td>
<td>Second condition in subclass and super-class patterns 1; synonymy exists cross-ontology</td>
<td></td>
<td></td>
<td>x</td>
</tr>
<tr>
<td>36 (&lt;relationWords, c_{jl}, synonymy, method &gt;)</td>
<td>String</td>
<td>Relation word contains second class name; e.g., (X hasParticipant Participant)</td>
<td></td>
<td></td>
<td>x</td>
</tr>
<tr>
<td>37 (&lt;relationWords, c_{jl}, synonymy, WordNet &gt;)</td>
<td>Linguistic</td>
<td>Relation word is synonym of second class name; e.g., (X speaker Person)</td>
<td></td>
<td></td>
<td>x</td>
</tr>
<tr>
<td>38 (&lt;relationWords, c_{jl}, synonymy, OpenCyc &gt;)</td>
<td>Semantic</td>
<td>Relation word is synonym of second class name; e.g., (X speaker Person)</td>
<td></td>
<td></td>
<td>x</td>
</tr>
<tr>
<td>39 (&lt;relationWords, c_{jl}, hyponymy, WordNet &gt;)</td>
<td>Linguistic</td>
<td>Relation word is hyponym of second class name; e.g., (X homeAddress Location)</td>
<td></td>
<td></td>
<td>x</td>
</tr>
<tr>
<td>40 (&lt;relationWords, c_{jl}, hyponymy, OpenCyc &gt;)</td>
<td>Semantic</td>
<td>Relation word is hyponym of second class name; e.g., (X homeAddress Location)</td>
<td></td>
<td></td>
<td>x</td>
</tr>
<tr>
<td>Evidence Primitive</td>
<td>Type</td>
<td>Explanation</td>
<td>Subclass</td>
<td>Superclass</td>
<td>Inferred Relations</td>
</tr>
<tr>
<td>-------------------</td>
<td>-----------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>----------</td>
<td>------------</td>
<td>--------------------</td>
</tr>
<tr>
<td>41</td>
<td>&lt; relationWords, c_{jl}, hypernymy, WordNet &gt;</td>
<td>Linguistic: Relation word is hypernym of second class name; e.g., (X speaker Person)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>42</td>
<td>&lt; relationWords, c_{jl}, hypernymy, OpenCyc &gt;</td>
<td>Semantic: Relation word is hypernym of second class name; e.g., (X speaker Person)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>43</td>
<td>&lt; c_{ik}, c_{jl}, sharesSynonyms, WordNet &gt;</td>
<td>Bag: Words in class labels contain synonyms; threshold of 2</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>44</td>
<td>&lt; c_{ik}, c_{jl}, sharesHyponyms, WordNet &gt;</td>
<td>Bag: Words in class labels contain hyponyms; threshold of 2</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>45</td>
<td>&lt; c_{ik}, c_{jl}, sharesHypernyms, WordNet &gt;</td>
<td>Bag: Words in class labels contain hypernyms; threshold of 2</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>46</td>
<td>&lt; c_{ik}, c_{jl}, sharesSynonyms, OpenCyc &gt;</td>
<td>Bag: Words in class labels contain synonyms; threshold of 2</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>47</td>
<td>&lt; c_{ik}, c_{jl}, sharesHyponyms, OpenCyc &gt;</td>
<td>Bag: Words in class labels contain hyponyms; threshold of 2</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>48</td>
<td>&lt; c_{ik}, c_{jl}, sharesHypernyms, OpenCyc &gt;</td>
<td>Bag: Words in class labels contain hypernyms; threshold of 2</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>49</td>
<td>&lt; c_{ik}, c_{jl}, sharesMeronyms, WordNet &gt;</td>
<td>Bag: Words in class labels contain meronyms (partOf relations); threshold of 2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>50</td>
<td>&lt; c_{ik}, c_{jl}, sharesHolonyms, WordNet &gt;</td>
<td>Bag: Words in class labels contain holonyms (hasPart relations); threshold of 2</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Bibliography


