

# VU Research Portal

## Using background knowledge in ontology matching

Aleksovski, Z.

2008

### **document version**

Publisher's PDF, also known as Version of record

[Link to publication in VU Research Portal](#)

### **citation for published version (APA)**

Aleksovski, Z. (2008). *Using background knowledge in ontology matching*. [PhD-Thesis - Research and graduation internal, Vrije Universiteit Amsterdam].

### **General rights**

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

### **Take down policy**

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

### **E-mail address:**

[vuresearchportal.ub@vu.nl](mailto:vuresearchportal.ub@vu.nl)



SIKS Dissertation Series No. 2008-24

The research reported in this thesis has been carried out under the auspices of SIKS, the Dutch Research School for Information and Knowledge Systems.

Cover illustration by Davor Iliev



VRIJE UNIVERSITEIT

# Using background knowledge in ontology matching

ACADEMISCH PROEFSCHRIFT

ter verkrijging van de graad Doctor aan  
de Vrije Universiteit Amsterdam,  
op gezag van de rector magnificus  
prof.dr. L.M. Bouter,  
in het openbaar te verdedigen  
ten overstaan van de promotiecommissie  
van de faculteit der Exacte Wetenschappen  
op dinsdag 24 juni 2008 om 10.45 uur  
in de aula van de universiteit,  
De Boelelaan 1105

door

**Zarko Aleksovski**

geboren te Skopje

promotor: prof.dr. F.A.H. van Harmelen  
copromotor: dr. W. ten Kate

# Contents

<b>1</b>	<b>Introduction</b>	<b>11</b>
1.1	Origin of ontology matching . . . . .	11
1.2	Need for ontology matching . . . . .	13
1.3	State-of-the-art matching tools . . . . .	15
1.4	Contribution of this thesis . . . . .	17
<b>2</b>	<b>Related work</b>	<b>19</b>
2.1	Matching ontologies directly . . . . .	20
2.1.1	Terminological matching techniques . . . . .	21
2.1.2	Structural matching techniques . . . . .	26
2.1.3	Instance-based methods . . . . .	29
2.1.4	Global methods . . . . .	30
2.2	Matching ontologies using external resources . . . . .	32
2.2.1	Background-based similarity measures . . . . .	32
2.2.2	Matching through a background ontology . . . . .	34
<b>3</b>	<b>Formal framework</b>	<b>39</b>
3.1	Using ontology as a background knowledge . . . . .	39
3.2	Terminology of the framework . . . . .	43
3.3	Evaluation of ontology matching . . . . .	47
3.3.1	Qualities of matching: Precision and Recall . . . . .	47
3.3.2	Matching Sample Evaluation . . . . .	50
3.4	Added value of background knowledge . . . . .	51

<b>4</b>	<b>Comparison with state-of-the-art matchers</b>	<b>53</b>
4.1	Introduction . . . . .	53
4.2	Case study description . . . . .	54
4.3	Baseline experiments . . . . .	57
4.3.1	Experiment 1: Lexical matching . . . . .	57
4.3.2	Experiment 2: Matching with state-of-the-art tools . . . . .	59
4.3.3	Discussion . . . . .	61
4.4	Experiment 3: Matching using background knowledge . . . . .	63
4.4.1	Anchoring . . . . .	63
4.4.2	Deriving relations . . . . .	66
4.5	Conclusions . . . . .	70
<b>5</b>	<b>Using multiple sources of background knowledge</b>	<b>71</b>
5.1	Introduction . . . . .	71
5.2	Case study 1 . . . . .	74
5.2.1	Experiments . . . . .	76
5.2.2	Evaluation . . . . .	79
5.2.3	Analysis . . . . .	81
5.2.4	Conclusions of case study 1 . . . . .	82
5.3	Case study 2 . . . . .	83
5.3.1	Experiments . . . . .	86
5.3.2	Evaluation . . . . .	89
5.3.3	Analysis . . . . .	89
5.3.4	Conclusions of case study 2 . . . . .	92
5.4	Conclusions . . . . .	93
<b>6</b>	<b>Combining background knowledge relations</b>	<b>95</b>
6.1	Introduction . . . . .	96
6.2	Combining relations in medical background knowledge . . . . .	97
6.2.1	Case study description . . . . .	98
6.2.2	Experiments . . . . .	99

6.2.3	Analysis of the results . . . . .	104
6.2.4	Lessons learnt . . . . .	110
6.3	Boost music search using domain background knowledge . . .	111
6.3.1	Music domain on the Internet . . . . .	112
6.3.2	Music test data extracted from Internet . . . . .	113
6.3.3	Music search using a background ontology . . . . .	115
6.3.4	Discussion . . . . .	119
6.4	Conclusions . . . . .	120
<b>7</b>	<b>Using unstructured background knowledge</b>	<b>123</b>
7.1	Introduction & motivation . . . . .	124
7.1.1	Introduction . . . . .	124
7.1.2	Internet Music Schemas . . . . .	124
7.2	An approximation method for ontology matching . . . . .	126
7.3	Heuristic weighting . . . . .	129
7.3.1	Defining weights . . . . .	129
7.3.2	Choosing Weights . . . . .	129
7.3.3	Normalised Google Distance . . . . .	131
7.3.4	Google-based weighting . . . . .	132
7.3.5	Modified Google Distance . . . . .	133
7.3.6	Examples . . . . .	134
7.3.7	Related work . . . . .	136
7.4	Experiments . . . . .	137
7.4.1	Experimental setup . . . . .	138
7.4.2	Comparing approximate and exact matching . . . . .	139
7.4.3	Google-based weighting experiments . . . . .	140
7.5	Conclusion . . . . .	142
<b>8</b>	<b>Conclusions</b>	<b>145</b>
8.1	Summary of main findings . . . . .	145
8.2	Discussion on success factors . . . . .	147
8.3	Future research directions . . . . .	150



8

*CONTENTS*

**9 Samenvatting**

**153**

# Preface

Artificial Intelligence is opening up a new era in our society. The question is no longer *if*, but *when*. Intelligent computers will become invisibly small and will enter every single bit of our lives. The way we live will drastically change. The world as we know it will be gone, and I can hardly wait to see what the new world will look like. This thought inspired me to do my PhD and write this book that You are now reading.

After four years of research, I realized that doing PhD in the Netherlands was very new and eye-opening experience for myself, well, I can not say a brilliant one, but there were some positive moments which I will remember long after this is over.

I want to thank to all my dear friends and colleagues who supported me during these four years: Aleksandar Pechkov, Vanja, Kiki, Riste, Robert, Dragan Sekulovski, Dragan Boshnjachki, Dragan Strashniot, Vano, Jasen, Meri, Davor, Peco, Cane, Toni, Ana, Anastasia, Bertran, Ogi, Hanriette, Maarten, Roni, Spyros, Gaston, Ruud, Mark, Clemens, Merlijn and finally Ilse Thompson. I want to thank my parents and my brother and his wife for their encouragement and support while I was far away from them. Especially I want to thank to my beloved Natasha, for the endless love and unconditional support when I mostly needed it, thank you.

Large part of this thesis is owed to my supervisors Frank van Harmelen and Warner ten Kate, who also invested a lot of time and resources to make this book happen.

I want to thank to the committee for reading and assessing my thesis. Finally, I want to thank Philips Research for funding the research presented in this thesis, and supporting me to make my thesis happen.

*So, I have decided to take my work back underground,  
to stop it falling into the wrong hands*  
(The Prodigy - from the album Music for the julted generation)



# Chapter 1

## Introduction

This chapter introduces the research topic of this thesis - *Using background knowledge in ontology matching*. Ontology matching has its roots in the knowledge representation and reasoning area. It is crucially important to knowledge-based information systems because many application scenarios need solid automatic techniques for ontology matching. The field attracted a lot of attention recently, and its active research community developed many prototype matching systems. This thesis is part of a recent trend to use background knowledge in the reasoning process.

To position the contribution of this thesis, we will briefly overview the historical origins and discuss the importance of ontology matching. We will refine the research topic to specific questions addressed in the rest of the thesis, and then we will summarize the major findings and the scientific contribution of this thesis.

### 1.1 Origin of ontology matching

*Knowledge Representation and Reasoning* explores different ways to organize, store and maintain knowledge in an information system in a way that will enable logical deductive reasoning, and consequently the execution of some "intelligent" task [Sowa, 2000, Davis et al., 1993]. The field has been and is constantly changing. Up until now people explored different formalisms to represent knowledge and reason about it.

*Semantic networks* [Woods, 1985] are based on the idea of human associative memory [Quillian, 1985]. They can be thought of as directed graphs

where the nodes represent concepts and the edges represent semantic relations between the concepts. They involve fairly loose semantic associations that are nonetheless useful for human browsing. Semantic networks have been used further to represent logic descriptions, such as *Existential Graphs* [Hammer, 1998] of Charles S. Peirce or *Conceptual Graphs* [Sowa, 1979] of John F. Sowa. Based on the Semantic Networks are also the so called *Frame-Based Systems* of Marvin Minsky [Minsky, 1985]. The idea of a frame system is rather simple: A frame represents an object or a concept. Attached to the frame is a collection of attributes, potentially filled with initial values. These values remain dynamic throughout the time and can be altered to correspond to a particular situation, therefore making possible to answer questions about the situation. Semantic networks also led to the early work on *Description Logics* [Baader et al., 2003] today better known after the introduction of KL-ONE [Brachman, 1977]. This work began with the goal of making term definitions in semantic networks more precise. Description Logics provide very expressive representation and reasoning languages with precise semantics. Sometimes not all of their expressiveness is needed, and weaker languages are used to enable the construction of automatic reasoners that can provide complete (and sound) inference in a tractable manner. Modern languages covering this spectrum of expressivity are *RDF*, *RDF-Schema*, *OWL*, *KIF*, *RIF(Rules)* and others.

At the present time ontologies are commonly used way to formally specify the knowledge in a domain. According to [Gruber, 1993], *an ontology is a formal specification of a domain conceptualization*. It distinguishes the different components that constitute the domain, and relates these components to one another to formally capture the knowledge of the domain. It also describes the vocabulary of that domain. The goal of such formalization is to create a machine processable model of the domain.

In the context of ontologies, knowledge modeling refers to the process of constructing an ontology as a conceptual representation of the domain knowledge. Typically a model will refer only to some aspects of the domain knowledge, and two models of the same domain may be essentially different, that is, the difference is more than just a simple renaming. This may be due to differing requirements of the ontology's end users or to conceptual or esthetic differences by the modelers and decisions made during the modeling process. This phenomena of generating different models for a single domain is known as heterogeneity in the ontologies. The problem to integrate such different ontologies which model the same or closely related domains is called *Ontology matching*. This problem is also known under various other names: *Ontology integration*, *Semantic integration*, *Ontology mapping*, *On-*

*tology alignment, etc.*

Current trends such as the Semantic Web [Berners-Lee et al., 2001], need automatic or semi-automatic solutions to the ontology matching problem. This can be either because the ontology size is too large to do the matching task manually, or because the ontologies evolve through the time and the matching is needed on a periodic basis. Depending on the application, different characteristics of the solutions are put forward as more important: quality, tractability, scalability, etc. [de Bruijn et al., 2004, Euzenat et al., 2004]. Lacking these solutions is a bottle-neck for the Semantic Web which envisions automatic knowledge sharing and manipulation on the Web. The importance of ontology matching is in the "sharing" part, namely, if no standardized interaction scheme exist, any form of combining ontology will first require a successful matching on the involved ontologies.

## 1.2 Need for ontology matching

The conflicts to be resolved in ontology matching are caused either by the use of different terminology or by differences between the conceptual models underlying the ontologies being matched. In some domains there is no ground truth, i.e. no authority exists that can confirm or disconfirm statements about the knowledge in the domain. A good example is the music domain where genres and styles are widely used to describe the music, but no agreement exists on the meaning of genres [Aucouturier and Pachet, 2003], we will discuss this example more extensively in Chapters 6 and 7.

As we mentioned above, depending on the task different matching qualities can be favoured in the solutions. When ontologies are integrated for the purpose of serendipity in human browsing<sup>1</sup>, for example, the integration is more important to be complete than precise - the user will be happy to see related subjects even if they are not exactly the ones he required. On the other hand, if the matching task is to reclassify patients from one to another hospital system, the diagnose must not be mistaken by any cost, and then the preciseness is much more favored over completeness. This shows that the success of ontology matching solutions very much depends on what the produced matches will be used for.

Ontology matching use cases are often discussed in the ontology matching literature, see [de Bruijn et al., 2004, H.Wache et al., 2001]. We will now

---

<sup>1</sup>The user browses one ontology and at the same time related subjects from another ontology are displayed along

describe some representative and well-known examples:

- *Web service integration.* The process of discovering web services can be understood as the process of finding a web service that fulfils a certain goal. The web service providers and the parties searching for such services can both describe their preferences and offered services using ontologies describing the domain of interest. Detecting the web service that corresponds to the profile described in the search preferences is actually matching the two, when a successful match is found the service can be proposed to the searching party as a possible solution.
- *Matching catalogues.* Many of the e-Commerce applications publish the goods and services they offer in an electronic catalogue. The users can purchase goods by selecting some of the items in the catalogue. In many scenarios, the online goods and service providers depend on successful integration of different catalogues.

Some of the online stores publish their catalogues in larger online e-mails which offer a variety of goods from various suppliers, and in order to do this, i.e. publish their offer, they have to integrate their offered items with the catalogues that the e-mail offers.

The alignment of different catalogues is a very challenging task because the goods and services are usually described through many different attributes, but they are of crucial importance to be accounted for in order for a successful integration to be performed.

- *Integrating multimedia resources.* A major challenge for multi-media archives is to offer efficient and intelligent information retrieval to the end users. The amount of multimedia content is vast and stored in digital collections which usually accompany the content with semantically different metadata. Currently it is believed that a good solution to the retrieval will need a successful integration of the semantically heterogeneous metadata annotations.
- *Peer-to-peer information sharing.* In Peer-to-peer usually there is no centralized authority that dictates the schema to be used by all peers in the network<sup>2</sup>. The network is unstable due to the possibility that different peers can join and leave the network at any time. The nature of peer-to-peer setting requires that the different peers interoperate

---

<sup>2</sup>Some existing networks do have a centralized schema to which all the users have to comply: eMule <http://www.emule-project.net/> uses a standardized schema for the content being exchanged throughout the network

and exchange information at run-time, and this requires the establishment of matches between their local schemas to enable the exchange. The success of such a scenario largely depends on the reliability of the ontology matching solutions to enable the information exchange.

- *Integrating document classifications.* Finding documents of interest in the overwhelming amount of available scientific publications is increasingly problematic. An attempt to reduce the problem is made by constructing different classification schemas, however, this still poses a big problem to the community in the sense that the retrieval of relevant documents is conditioned by having a unified access to all the different classifications at once. The semantic integration problem is two-fold in this setting: the classifications need to be integrated to provide a unified view to their content, and this integrated view can be further combined with the semantic analysis of the document's content to successfully identify the documents of interest. Among the others, an example of a large-scale integration effort is UMLS, which counts the indexed entities in millions [Bodenreider, 2004].

### 1.3 State-of-the-art matching tools

State-of-the-art matching tools combine multiple approaches to produce a matching between ontologies. The main components are usually lexical and structural matching techniques. The lexical techniques detect similarities between the elements using their names and descriptions, and the structural exploit the information of how the entities are structured within the matching ontologies to find the similarities. In this section we do not aim to give an exhaustive state-of-the-art survey. Instead, we briefly point to some of the main players in the field. An exhaustive and up-to-date survey can be found in [Euzenat and Shvaiko, 2007].

- *Prompt* [Noy and Musen, 2000] is a tool that offers a semi-automatic solution to the ontology matching problem. Prompt executes some tasks automatically, and based on the results interacts with the user when user's assistance is needed. When performing the automatic matching part, or after taking an input from the user it checks for inconsistencies and suggests solutions to the user for these problems.

Prompt takes two ontologies as input, and produces one merged ontology as a result. It operates iteratively, first suggesting a list of matches,



and then giving the user the possibility to either change some of the suggested matches or introduce new one. After the user's intervention, it recalculates new suggested matches taking into account all the previous user actions. The process ends when the user is satisfied with the produced resulting matches.

- *Cupid* [Madhavan et al., 2001a] is a generic schema matching tool that discovers mappings between schema elements based on their names, data types, constraints, and schema structure. It makes integrated use of linguistic and structural matching, context-dependent matching of shared types, and it is biased toward leaf structure where much of the schema content resides.

It uses automated linguistic-based matching component to find correspondences between the elements in the schemas. It considers the similarity of atomic elements (i.e. leaves in the schema) as crucial in the matching, assuming that most of the semantics is captured in them. It is able to find context-dependent matches which is used in different schemas.

- *Coma++* [Aumueller et al., 2005] is a schema matching system developed with a purpose to provide a platform that can combine multiple matching techniques in a flexible way. The framework offers the possibility to flexibly combine different matchers, and also enables the results of one applied technique to be reused in another matcher. The experience of using COMA, according to [Do and Rahm, 2002], shows performance superiority of combining matching approaches, and indicates a high value of reusing matches.
- *HMatch* [Euzenat et al., 2006] is a system which intends to find semantically corresponding concepts from two ontologies with the same or closest meaning. The semantic closeness is calculated as a compound value of lexical similarity which uses Wordnet for lexical variations, and structural similarity which accounts for the context of the concept in the ontology. The concept's context include its properties, property values and relations to the other concepts in the ontology. The semantic closeness is represented as a value in the interval  $[0, 1]$  and the matches are produced by setting a threshold which when surpassed the system will report a match between the comparing concepts.
- *RiMOM* [Euzenat et al., 2006] is a tool for ontology alignment which combines three different strategies to produce an ontology matching. It uses a lexical matching component based on edit-distance similarity

(see Chapter 2 for description of this technique), statistical learning component, and a structural component based on similarity propagation among concepts and properties.

The tools *Falcon-AO* and *FOAM* were used in our experiments, and we will describe them later in Chapter 4. *Falcon-AO* is well known by its success in the Ontology Alignment Evaluation Initiative, it achieved the best performance results in OAEI 2005 and 2006. Other sources for an overview of the tools being currently under active development are [Euzenat et al., 2006, Euzenat et al., 2007].

## 1.4 Contribution of this thesis

This thesis contributes to the research topic of using background knowledge in ontology matching. The research was conducted using real-life data from different domains: medical, music and agriculture. The main question this thesis aims to answer is:

**How can background knowledge be used in ontology matching?**

To provide an answer, we distinguish more specific research questions addressing different aspects of the topic:

- **1. Can the use of background knowledge contribute to the state-of-the-art ontology matching tools?**

The use of background knowledge can contribute to the state-of-the-art ontology matching tools. In Chapter 4 we compare the use of background knowledge against the performance of the state-of-the-art ontology matching tools *Falcon-AO* and *FOAM*. We conducted the research on a case study which is representative of many realistic matching cases, and we involved domain experts in the process of assessing the matching performance.

- **2. What is the benefit of using *multiple ontologies* as background knowledge simultaneously?**

In Chapter 5 we show that the cumulative use of multiple background knowledge ontologies can result in better performance than using any of them alone. When adding more background ontologies, the recall increases considerably in a monotonic manner while the precision changes depending on the quality of the background knowledge - high quality results in high precision and low quality results in low matching precision.

- **3. How does a *combination of different domain relations* in the background knowledge contribute to the matching performance?**

In Chapter 6 we show that combining relations of different types within the background ontology is more beneficial than using relations of a single type. Relations of different types need careful consideration on how they can be combined, but the analysis shows that even relation types of very different nature can be successfully combined.

- **4. How flexible is the scheme of using background knowledge in ontology matching?**

In Chapter 7 we show, first, that background knowledge can be applied not only to discover ontology matches but also to improve an approximate matching discovery, and second, as background knowledge we can also use other sources of knowledge - in our case it is the Google search engine. This way as a background knowledge we make use of the huge amount of knowledge that is implicit in the current Web.

Using background knowledge in ontology matching is practically feasible and useful. It provides a clear contribution to the existing matching techniques. Importantly, it shows the following desirable properties: it is feasible to combine multiple background ontologies for a cumulative benefit, it is useful to combine different domain specific relations within the background knowledge, and it is feasible to use an unstructured source of information as background knowledge instead of the structured ontologies.

In the next chapter we will provide a survey of the related work which can be found in the literature on ontology matching. We will focus on the essential techniques which are employed in the matching tools. In chapter 3 we will describe a formal framework which we use in the rest of the thesis. Finally, in Chapter 8 we will summarize all the findings of this thesis and discuss the important research directions for the future.

# Chapter 2

## Related work

The ontology matching problem is known to the artificial intelligence research community for the past few decades. It was known under various names: semantic integration, ontology integration, ontology alignment, ontology matching, etc. First recognized in the field of knowledge representation and reasoning, the problem was identified when knowledge bases were developed that could benefit from one another. Such knowledge bases would for example describe the same or very related domains. Realizing this possible mutual benefit requires their integration, that is establishing a kind of bridge between them.

The state of the art on ontology matching is not restricted to a single discipline, in the database community much work has been done on integrating schema and instance data from heterogeneous databases. The distinction between databases and ontologies can be seen in the definition of semantics. The DL-based OWL ontologies, for instance, use open-world, whereas relational databases use close world semantics. Good recent comparison is presented in [Motik et al., 2007]. Furthermore, database matching is usually restricted to the traditional relational model [Codd, 1970], while ontology matching uses a richer data-model such as hierarchical (e.g. RDF Schema), or even richer like description logic. Yet, as discussed in [de Bruijn et al., 2004], these distinctions are minor as regarded to the scope of integration, making it possible to consider the work done on schema matching in the database area as a form of ontology matching. Good sources for an overview of database schema integration are [Roddick et al., 2003, McCallum et al., 2000].

In this thesis we discuss the problem of ontology matching, emphasizing the paradigm of using background knowledge. For the purpose we divide the related work in two categories: work on direct ontology matching and work on

using external resources in ontology matching. Other categorization criteria have been discussed elsewhere [Noy and Musen, 2000, de Bruijn et al., 2004, Euzenat et al., 2004, Shvaiko and Euzenat, 2005], and include: the representational complexity of the matching ontologies, the type of descriptions being matched, element-level v.s. structure-level, etc.

This division on direct matching, and using external resources in the matching is more of a gradual spectrum than a strict division. Many techniques fall somewhere on the spectrum between these two extremes. When matching simple vocabularies, for example, one approach is to apply a generic edit distance type of metric. Depending on the language of the ontologies (that is the language used to name the entities in the ontology - English, German, etc.), some edits can happen more often than others. Having this language-specific information, one can modify the edit distance in such a way that it matches more accurately when used for that particular language. Similarly, domain specific knowledge can be used in the design of the matching algorithm itself. In some germanic languages words are often compounded together to form a new word - *Hersen Tumor* is written as *Hersentumor* (which means brain tumor in the Dutch language). Accounting for these constructions can boost the matching performance, [Aleksovski et al., 2006a]. Stemming [Hull, 1996] is yet another well-known example from Natural Language Processing (NLP) which uses the domain knowledge for a particular language to detect different word variations coming from the same origin.

In line with the given division, in the rest of this chapter we will first summarize the direct ontology matching techniques, and then review the techniques which use explicit external knowledge.

## 2.1 Matching ontologies directly

Techniques for matching ontologies directly use solely the information contained in the two ontologies being matched. This information depends on the actual ontologies, and comprises: text descriptions, relatedness of the entities (structure), topology of the structure, classified instance data, etc.

Different criteria can be applied to classify a direct matching technique. They include the general properties of the algorithm, the interpretation of the input information, the kind of input information, the expected output of the matching etc. More detailed discussions on this issue can be found in [Noy and Musen, 2000, de Bruijn et al., 2004, Rahm and Bernstein, 2001, Shvaiko and Euzenat, 2005, H.Wache et al., 2001]. However, the matching

techniques naturally cluster in four categories: terminological, structural, instance-based and global matching techniques. In this overview we use this categorization.

Each of the four categories can be subject to separate analysis and finer-grained division, but they are not entirely independent of each other. Some use the output of others as their own input. The structural techniques, for instance, are often considered not reliable when used alone, but combining them with the terminological matchers can result in a good matching performance.

Now, we introduce the formal notions that we will use in describing the techniques.

- *String* is a finite sequence of characters. Strings are referred to with non-capital italic alphabetic letters, and most often we will use  $x$  and  $y$  to refer to the two matching strings under scrutiny.
- $\mathbb{S}$  is the set of all strings.
- $|x|$  denotes the length of the string  $x$ .
- $x[i]$  is the  $i$ -th character in the string.  $i$  ranges from 1 to  $|x|$ .
- $x + y$  is the concatenation of the two strings  $x$  and  $y$ , i.e.,  $(x + y)[i] = x[i], i = 1..|x|$  and  $(x + y)[i] = y[i - |x|], i = |x| + 1..|x| + |y|$ .

### 2.1.1 Terminological matching techniques

Terminological methods compare strings. They match the text descriptions assigned to the elements in the ontologies like names, their alternative synonyms, comments, or identifiers like URI or URL. Terminological methods can operate with the text descriptions regarding them as simple arrays of characters, and also interpreting them as a meaningful text. The later use language-specific knowledge to detect different word forms or interpret meaningful constructions of word sequences like noun phrases, sentences, paragraphs etc.

**Term normalization.** Before terms are compared, they are usually normalized first. Among the numerous normalization procedures, some are:

- case standardization (convert all the alphabetic characters to lower case) to avoid case sensitivity in the comparison,

- elimination of multiple blank spaces in between words,
- diacritic suppression, that is replacing characters with diacritic signs with their most frequent replacement (for example replace à with a),
- punctuation elimination,
- digit suppression,
- stop-word elimination to discard words like *the*, *of*, *a*, when longer labels or whole pieces of text are compared, etc.

The normalization compensates for syntactical differences which are trivial for humans (in particular the language natives), but represent serious problem when automatic comparison is performed.

The terminological techniques produce a number as a result. It represents some kind of closeness detected between the comparing strings, and can have two scales: similarity and dissimilarity measure. In the first case a technique returns 0 when the comparing strings are different and other positive value when they are found equivalent. The maximal similarity value is obtained when a string is compared with itself (and this value is usually 1). Analogously, dissimilarity returns 0 when the strings are similar, and other positive value when they are dissimilar. That positive value characterizes the degree of dissimilarity between the comparing strings. Often, normalization to the result is applied. It can be used, for example, to bound the dissimilarity value to a maximum of 1, or can compensate for variable lengths of the compared strings.

### String-based matchers.

String-based matchers calculate the similarity between two terms solely based on the strings representing the terms. Since this is a generic method, string-based matchers are widely applicable. However, their efficiency varies in different contexts because they lack any semantic account of the text being matched. Some commonly used similarity and dissimilarity metrics are as follows:

- **String equality** is a similarity measure  $\delta : \mathbb{S} \times \mathbb{S} \rightarrow \{0, 1\}$  such that  $\forall x, y \in \mathbb{S} \delta(x, y) = 1$  if  $x = y$ , and  $\delta(x, y) = 0$  if  $x \neq y$ .

In this form the measure is rigid, and its effectiveness is highly increased when normalization is applied first. Used without normalization it

would, for example, not detect that *Lion* and *lion* actually refers to the same thing.

When comparing two strings this measure determines whether they are equal or not, and in case the strings are different it does not say anything about how much they are different. The measures that follow do provide information about this difference.

- **Substring** is a similarity measure  $\delta : \mathbb{S} \times \mathbb{S} \rightarrow \{0, 1\}$  such that  $\forall x, y \in \mathbb{S} \delta(x, y) = 1$  if  $x = p + y + q$  or  $y = p + x + q$  for some  $p, q \in \mathbb{S}$ , and  $\delta(x, y) = 0$  otherwise.

The substring measure is more relaxed than the equality measure. Terms like *Lion* and *Lions* will be reported similar by substring, but not by the string equality measure. On the other hand, *ion* and *Lion* will also be reported similar. Each pair of strings detected by the string equality will be detected with the substring measure as well. However, substring similarity has to be used carefully because the strings found similar can be very different in size, and consequently in their meaning. Used in specific contexts it has proven useful, [Aleksovski et al., 2006a].

- **Hamming distance** is a dissimilarity measure  $\delta : \mathbb{S} \times \mathbb{S} \rightarrow \mathbb{R}$  such that:

$$\delta(x, y) = |\{i : x[i] \neq y[i]\}|$$

The Hamming distance between two strings of equal length is the number of positions for which the corresponding characters are different. Put another way, it measures the number of character substitutions required to change one string into the other. When the strings are different in length, the measure results in a value not lower than the length difference. This measure is inspired by information theory, in the area of error detection and error correction codes, see [Hamming, 1950].

- **N-gram distance** between two strings counts the number of common substrings of equal length. Let  $Sub(x, n)$  be the set of substrings of  $x$  of length  $n$ , then the n-gram distance is a similarity measure such that:

$$\delta_n(x, y) = |\{Sub(x, n) \cap Sub(y, n)\}|$$

An n-gram of length 1 is called unigram, length 2 bigram, etc. Usually this measure is normalized with the string length, therefore becoming a relative measure for similarity.



- **Edit distance (Levenshtein distance)** between two strings is determined by the minimum number of operations needed to transform one string to the other. The operators are restricted to affect a *single* character in the string. Given a set of operations  $O = \{op : \mathbb{S} \times \mathbb{S}\}$ , the edit distance is the dissimilarity measure  $\delta : \mathbb{S} \times \mathbb{S} \rightarrow \mathbb{R}$  such that  $\delta(x, y)$  is the minimal length of sequence of operations that transforms  $x$  in  $y$ .

$$\delta(x, y) = \min(\{n : (\exists op_1, op_2, \dots, op_n \in O)(op_1(op_2(\dots op_n(x)\dots))) = y\})$$

The basic version published in [Levenshtein, 1966] accounts for substitution, insertion and deletion of a single character as valid operations.

Levenshtein edit distance is probably the most widely used approximate string distance measure, with many extensions and variations. Some of them assign a cost function to each operation and then minimize the distance, that is the sum of all separate cost values for the operations in the sequence. The cost function of a single operation may produce different values depending on the characters involved in the transformation. Typically the variations of this distance measure alter the cost function that is assigned to the operations. The more known variations of this measure include:

- **Damerau-Levenshtein distance** is an extension of the Levenshtein distance that counts transposition as a single edit operation, rather than two. Transposition means exchanging two adjacent characters in a string. This measure is therefore the minimal number of insertions, deletions, substitutions and transpositions needed to transform one string to another. In his seminal paper [Damerau, 1964], Damerau stated that these four operations correspond to more than 80% of all human misspellings. He proposed this distance measure as a solution to the misspelling problem.
- **Needleman-Wunch distance** is the edit distance with a higher costs for insertion and deletion of character. It was proposed in [Needleman and Wunsch, 1970] and is commonly used nowadays in bioinformatics for the alignment of amino acid and nucleotide sequences. In their terminology the insertion and deletion operations are called gaps in the transformation. In the context of these sequences it is known that the gaps are much less likely to occur, and therefore they are assigned high penalty in the comparison.
- **Smith-Waterman distance** is described in [Smith and Waterman, 1981], allows different values for the cost function of a single

operation. It was developed to identify optimal alignments between related DNA and protein (peptide) sequences.

- **Jaro distance**, first proposed in [Jaro, 1995], is a similarity measure based on the number and order of common characters between the comparing strings. It is a dissimilarity measure  $\delta : \mathbb{S} \times \mathbb{S} \rightarrow \mathbb{R}$  defined as follows:

$$\delta(x, y) = \frac{m}{3|x|} + \frac{m}{3|y|} + \frac{m - t}{3m}$$

where

- $m$  is the number of "matching" characters from  $x$ , meaning that the same character appears in  $y$  on a position not farther than  $\frac{\min(|x|, |y|)}{2} - 1$  from its position in  $x$ .
- $t$  is the number of transpositions. This number is calculated by counting the matching characters in  $x$  different from the character in the same position in  $y$ , and then divided by two. Intuitively, the transpositions can be understood as the matching characters which are out of order.

**Jaro-Winkler distance** is one known variation of the original Jaro distance. It uses a prefix scale which favors the ratings to strings that match from the beginning for a predefined prefix length. Details can be found in [Winkler, 1999].

### Language based terminological matchers.

Language based matchers constitute the other main group of term-based methods. They assume that the ontology items have been given meaningful names or descriptions, and these are in the same language. They first perform morphological and syntactic analysis to normalize the terms, and subsequently match the normalized terms. They rely on the expressive and productive properties of the natural languages, meaning that the terms can be expressed in many different ways without intrinsically altering their semantics [Diana and Anaiadou, 1999]. This group of algorithms in the literature is positioned in the area of Natural Language Processing. A couple of the most famous include:

- **Stemming.** Morphological variants are most commonly identified through stemming algorithms. They trim words to their base form

by removing suffixes such as plural forms and affixes denoting declension or conjugation. For example, *cut*, *cutting*, *cuts* would be reduced to the single stem *cut*, making it trivial to recognize the same meaning behind.

- **Different word forms.** Besides their normal form, words, noun phrases and other expressions can be referred to using other forms which evolve through time. Such forms are the short word forms, with one specific type of short forms called acronyms. Examples of short forms are: *NRG* which when spelled sounds very similar to the pronunciation of *Energy* which is the actual intended meaning. The short form *A'dam* stands for *Amsterdam* which is a town in the Netherlands. Acronym is an abbreviation of a long noun phrase that consists of multiple words. The short form then consists only of the first or representative letters of each word in the noun phrase. For example, *FTE* means *Full Time Equivalent*, *XML* means *eXtended Markup Language* and *KPN* means *Koninklijke PTT Nederland* which even contains a nested acronym within the noun phrase. The new culture that evolves with online chatting and SMS in the mobile phone world has enriched this language set considerably.
- **Token-based distance.** The token-based similarity measures are usually applied to compare whole pieces of text, not only names. First, the text is segmented into tokens which are generally substrings of the text, and then the pieces of text are compared treating them as sets of strings (sets of tokens). When the frequency of occurrence of each token is taken into account, the comparison becomes comparing multi-sets of strings. They are regarded as vectors with each dimension corresponding to a token, and the value to its frequency of occurrence. Several versions of the Jaccard similarity, which we elaborate in 2.1.3, have been used in this context to compare the vectors as an intermediate step in comparing the texts. The performance of some of them is reported in [Bilenko et al., 2003, Cohen et al., 2003].

## 2.1.2 Structural matching techniques

Structural matching techniques use the underlying structure of the ontologies to perform the matching. An ontology in this context is regarded as a graph of interconnected entities, and the main idea behind this matching is that similar entities are likely to exhibit similar semantic neighbors (that is the

other entities in the ontology that they relate to). This information is used in several ways.

The graph can be treated as a purely topological structure, or as a hierarchical order. In the latter case the relation posing the hierarchical order is given a special treatment, since hierarchical relations are usually transitive.

**Graph matching.** In this approach a similarity measure is computed based on a topological correspondence between the two graphs. No special treatment is given to the different relations assigned to the edges in the graph. This viewpoint is taken in [Hu et al., 2005], where the structure of the ontologies is modeled as a bipartite graph. Similarity of two entities is calculated as accumulation of similarities of their involved relations (triples - two entities connected with a relation), and the similarity of relations is calculated as accumulation of similarities of their involved entities.

First, matched pairs of entities are found, typically by using some terminological matcher. This part is taken as input to produce additional matching pairs by comparing the structural similarity around the given pair. The method does not depend on lexical similarity per se, however, it has been reported that purely structural similarity is not accurate.

**Hierarchical matching.** In this case the hierarchical order of entities in the ontology is taken into account. Hierarchical order stems from transitive relation, and is therefore a partial order, i.e. Directed Acyclic Graph rather than an arbitrary graph form. Two main types of hierarchical order are the mereological and taxonomical.

- **Mereological hierarchies** connect the entities through *part-whole* relations. An entity lower in the hierarchy *is part of* any entity higher in the hierarchy to which it relates. Vice versa, the entity higher is composed of different parts which are its relating entities lower in the hierarchy. This generic relation is used in a wide range of domains: in geography to organize countries, regions, cities, rivers, etc.; in chemistry to express inclusion of substances (sugar *is part of* honey); in anatomy to effectively organize the vast amount of different concepts (hand *is part of* arm); and so on.

In some ontologies the part-whole relations are further refined for more accurate modeling. In the Foundational Model of Anatomy (FMA), for example, three types of *part-whole* relations are distinguished: *constitutional part-of*, *regional part-of* and *systemic part-of* [Rose and Jr.,

2003]. On top, attributes are attached to each relation to capture further details.

A method to discover *part-whole* relations was reported in [van Hage et al., 2006]. The study distinguishes six different kinds of *part-whole* relations. They are discovered in two steps: first, phrase patterns that encode the appropriate semantics of the particular kind of *part-whole* are discovered, and second, they are applied by searching through documents to find concrete instances of the relation.

- **Taxonomic hierarchies** connect the entities through a parent-child relationship. Taxonomy refers to a classification of things<sup>1</sup>, but in principle they are structured hierarchically.

In matching systems several different measures have been used to exploit the taxonomical structure. One such a measure is proposed in [Valtchev and Euzenat, 1997]. It is called structural topological dissimilarity on a domain. It measures the semantic distance as the shortest path between two nodes in the hierarchy. Intuitively, if the shortest path passes through entities high in the taxonomy, the semantic connection is weak and the entities are concluded dissimilar.

The hierarchical order of entities is used in matching mainly through several intuitive assumptions. Two entities can be considered similar if:

- their super-entities are similar,
- their sub-entities are similar,
- majority of their sub-entities are similar,
- majority of their super-entities are similar,
- all of their leaf entities are similar.

When applied in an actual matching system best performance is obtained when the above intuitions are combined in a sensible manner, see [Madhavan et al., 2001b].

---

<sup>1</sup>Originally only for classifying organisms, <http://www.cmdr.ubc.ca/pathogenomics/terminology.html>

### 2.1.3 Instance-based methods

Instance-based matching uses the instances classified in the two ontologies being matched. The similarity between two classes is assessed by observing the number of instances that these classes share, their individual number of instances, and the total number of instances they contain. The most frequently used metrics are:

- **Jaccard similarity.** The Jaccard similarity between two sets  $X$  and  $Y$  is defined as the ratio between the size of the intersection and the size of the union of the two sets being compared:

$$\delta(X, Y) = \frac{|X \cap Y|}{|X \cup Y|}$$

where  $|\cdot|$  denotes the size, i.e., the number of elements in the set.  $\delta(X, Y)$  equals 0 when the sets are disjoint and equals 1 when they are identical. This is very intuitive measure for the given setting as it measures the relative overlap of the two sets. There is also a probabilistic interpretation, namely the probability that a random instance from the union is in the intersection of the two sets.

- **K-Statistic.** A method to determine similar categories of documents from directory structures is proposed in [Ichise et al., 2003]. It adopts the so-called K-Statistic method for evaluating the similarity between the categories. The method exclusively relies on the categorization structure of the directories, i.e., without any reliance on the semantic information of the documents. The categories are thought of as classification criteria that determine whether a document should be classified into the category or not. Then for two comparing categories from the directories  $C_s$  and  $C_t$ , four numbers are calculated:

- $N_{11}$  - the num. of doc. belonging to both  $C_s$  and  $C_t$
- $N_{12}$  - the num. of doc. belonging to  $C_s$  but not  $C_t$
- $N_{21}$  - the num. of doc. not belonging to  $C_s$  but belonging to  $C_t$
- $N_{22}$  - the num. of doc. not belonging to any of  $C_s$  and  $C_t$

Based on these numbers two probabilities are calculated: One is the observed occurrence of instances that are both in or both out of  $C_s$  and  $C_t$ , i.e. the normalized form of  $N_{11} + N_{22}$ , and one on the coincidental occurrence of instances being both in or out, estimated by the normalized form of  $|C_s| |C_t| + |C_s^c| |C_t^c|$  (where  $C_s^c$  denotes the complement

set of  $C_s$  in terms of instances). Then, the value of  $k$ -statistic is calculated based on these two probabilities, and is consequently used to estimate with a certain significance the similarity of the two concepts. For further background in statistics see [Fleiss et al., 1981].

Difficult cases arise when there are no shared instances between the matching ontologies. Then it is still useful to compute some kind of distance between the classes, even though the above-mentioned methods will find no match. There are methods that compare classes based on the mutual similarities between their instances. These similarities can be assessed using any of the matchers presented before. Some of these methods are described in [de Bruijn et al., 2004].

### 2.1.4 Global methods

Global methods compute matching between two ontologies, taking one or two matching results produced by other methods as input. The resulting matching of these methods optimizes for certain criteria. For instance, some methods compute many-to-many matches while the required result is one-to-one matching<sup>2</sup>. A global method would typically select a subset of these many-to-many matches that satisfies the constraint and is optimal in some respect.

Among others, global methods include aggregating the results of different matchers, iterative behavior of matchings, involving the user in the matching process, etc. We discuss three representative examples of global methods.

- **Weighted sum.** Weighted sum computes a similarity measure between two classes using several other similarity measures as input. The final similarity is composed as a weighted sum of the input measures. The weighted sum between the classes  $X$  and  $Y$  given  $n$  different similarity measures  $\delta^i$  with assigned weights  $w_i \in (0, 1)$ , is calculated as follows:

$$\delta(X, Y) = \sum_{i=1}^n w_i \cdot \delta^i(X, Y)$$

where

$$\sum w_i = 1$$

---

<sup>2</sup>One-to-one matching means each entity can be matched to at most one in the other ontology.

ensuring that the compounded value is a similarity measure again. The weights are depictive of the "authority" that is given to each  $\delta^i$  measure. Higher weight means higher authority. Due to this intuition, weighted sum is sometimes called a jury system.

- **Improving Automatically Created Mappings Using Logical Reasoning.** In [Meilicke et al., 2006], the authors describe a collection of methods for improving an ontology matching result that contains wrong or redundant matches. It takes the set of matches as input and produces an improved result set using logical reasoning in the context of Distributed Description Logics [Borgida and Serafini, 2003]. The authors refer to this process as debugging of matches.

The debugging uses the notion of consistency of matches to detect problems. Simple example of a problem is as follows: Two ontologies  $O'$  and  $O''$  being matched, contain the following axioms:

$$O' : Author \sqsubseteq Person$$

$$O'' : Person \sqsubseteq \neg Authorization$$

Lexical matcher finds the resulting set:

$$O' : Person \equiv O'' : Person$$

$$O' : Author \equiv O'' : Authorization$$

and now, using DDL reasoning an inconsistency is detected. It can be understood as an inconsistency that occurs when all the axioms and matches in the case are considered as axioms of a single ontology. The debugging process is semi-automatic, and uses heuristic algorithm to propose which matches are the likely candidates to be discarded.

- **User feedback.** The user can be involved in the matching loop combining his feedback with other techniques iteratively. Every time the algorithm gets new user input it recalculates an improved output. Positive experience is reported from this approach using the matching tool Prompt, see [Noy and Musen, 2000].

In the survey [de Bruijn et al., 2004] machine learning techniques applied in ontology matching are classified as global methods. In this case the matcher is trained on a set of examples after which it is used to match new cases.



## 2.2 Matching ontologies using external resources

Traditional ontology matching approaches are restricted to the use of the information available in the ontologies being matched. Some of the new techniques go beyond this, and use external background knowledge in the matching. This background knowledge is used in different ways and from various kinds of knowledge sources. In the rest of the text by using background knowledge, we will refer to *automatic exploitation of an external source of knowledge*.

There are two main aspects to be considered: relating the ontologies being matched to the background knowledge and using the knowledge provided by that background source. The precise way this is done affects the outcome of the matching process. We distinguish two categories of using background knowledge: background-based similarity measures and matching through a background ontology.

### 2.2.1 Background-based similarity measures

Background-based similarity measures use external resources as an oracle (authority) to determine the similarity between two entities from the two ontologies being matched. The two entities are provided to the oracle which returns a similarity ranking/rating. It is assumed that the terms are known to the oracle, or somehow can be matched (anchored) to entries stored in the oracle. To use external resource in finding a similarity measure, the matching entities have to be anchored to the external resource in some way.

- **Synonymy detection.** Synonymity detection means finding pairs of words which have the same meaning. For example *Human* and *Person* can be considered synonymous words. Detecting synonyms can not be done with simple syntactic or lexical rules, but instead the techniques to find them require external resources where the synonyms can be retrieved from. Such external resource can be a vocabulary, a thesaurus, an ontology etc. Exploiting synonymity is a basic form of using external resources. Most matching tools employ a synonym list.
- **Resnik similarity** A measure more elaborate than synonymity is the one proposed in [Resnik, 1999]. In a language, a term can have multiple meanings, which altogether are considered as a set of meanings - a synset. The Resnik similarity measure takes into account that the terms can be part of several synsets and exploits the is-a hierarchy

between the synsets. Each synset  $C$  is associated with a probability of occurrence  $P(C)$  of an instance of the concept associated to a particular synset. This probability is obtained from the text corpus using the occurrence frequency. This measure draws on the assumption that terms with more specific meaning will occur less often and hence will have a lower probability. Similarly, high frequency terms have a low information content. The similarity between two terms is a function of the common synset of both terms which maximizes the information content. This common synset is found climbing up the *is-a* hierarchy. For two different concepts in a taxonomy, this similarity measure can be understood as the information of the most informative subsumer of the two concepts.

Experiments conducted using Wordnet as an external resource revealed that this similarity measure outperforms the similarity measure based on the pure path connection length of concepts and the probability of co-occurrence.

- **Multilingual methods** Multilingual methods are about matching terms from different languages. They would typically do the translation using a multilingual dictionary such as EuroWordNet<sup>3</sup>, although there are other possible methods. They use techniques from machine translation and classical ontology alignment methods that assume single-language ontologies [Fung and Yee, 1998]. Much work is reported on semantic matching within a single language, but very little on cross-lingual semantic matching. Aside from the problem of a good quality matching within a single language, one of the difficulties that cross-lingual matching poses, is that there can be many-to-many translations of words, which then requires the use of some kind of similarity measure.

A project aiming at creating a multilingual lexicon for Dutch, English and German was Polylex<sup>4</sup>. The lexicon was designed in such a way that it allows multiple inheritance. There is one common hierarchy and there are separate hierarchies for each language specifically. This allows the distinction between language specific relations.

- **Normalized Google distance as dissimilarity measure.** Normalized Google Distance (NGD) dissimilarity measure was introduced in [Cilibrasi and Vitanyi, 2007]. It takes advantage of the number of hits

---

<sup>3</sup><http://www.illc.uva.nl/EuroWordNet/>

<sup>4</sup><http://www.informatics.susx.ac.uk/research/nlp/polylex/polylex.html>

returned by Google on a query. The names (labels) of the concepts to be compared are fed to the Google search engine as search terms. Given two search terms  $x$  and  $y$ , the Normalized Google Distance between  $x$  and  $y$ ,  $NGD(x, y)$ , is obtained as follows:

$$\delta(x, y) = \frac{\max\{\log f(x), \log f(y)\} - \log f(x, y)}{\log M - \min\{\log f(x), \log f(y)\}}$$

where  $f(x)$  is the number of Google hits for the search term  $x$ ,  $f(y)$  is the number of Google hits for the search term  $y$ ,  $f(x, y)$  is the number of Google hits for the tuple of search terms  $x + y$ , and  $M$  is the number of web pages indexed by Google. Intuitively,  $NGD(x, y)$  is a measure for the symmetric conditional probability of co-occurrence of the terms  $x$  and  $y$ : given a web-page containing one of the terms  $x$  or  $y$ ,  $NGD(x, y)$  measures the probability of that web-page also containing the other term.

## 2.2.2 Matching through a background ontology

By matching through a background knowledge ontology we refer to the application of reasoning mechanisms over the background knowledge to deduce matches between provided entities. This type of matching is different from the background-based similarity measures discussed in the previous section in the sense that those similarity measures do not explain how the entities are similar. Matching through a background ontology explicitly involves reasoning and therefore is able to also find relations other than "hard-coded" equivalences. They can, for example, discover that one entity is more general or less general than another. In the literature several approaches have been proposed to realize matching through a background ontology.

- **Ctx-match matcher.** Ctx-match [Magnini et al., 2004, Bouquet et al., 2003] is an algorithm for automatically discovering relations across ontologies structured as hierarchies, and used for classification purposes (simply called classification hierarchies). The problem of discovering the relations is encoded as a problem of logical satisfiability, and the discovered matches have a well-defined semantics. The algorithm has two main phases: semantic explicitation and semantic comparison.

In the first phase, semantic explicitation, the concepts from the matching ontologies are interpreted into the background knowledge ontology,

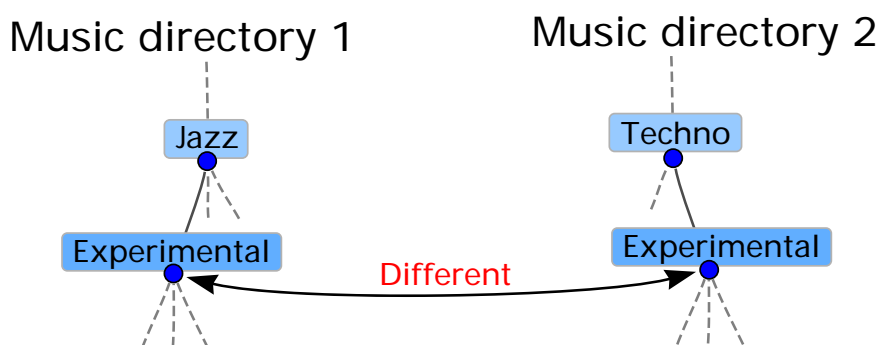


Figure 2.1: Ctx-Match finds that two identically named classes have different meaning due to the context in which they appear.

such that each concept is associated with a logical formula built from the background knowledge concepts. Semantic explicitation creates an explicit expression that captures the meaning of a concept in the form of a logical formula. It is done in two steps: linguistic interpretation and contextualization. In the first step a terminological match is performed to find corresponding concepts in the background knowledge ontology, and in the second step the semantic neighborhood of the concept is encoded in the formula by adding its parents as conjuncts.

In the second phase, semantic comparison, the matching concepts are compared by comparing their explicitated formulas. From the background ontology a set of relations is extracted and used as an axiom set in the further comparison. SAT solver is used to find the match between two concepts based on the set of axioms previously extracted.

The Ctx-match approach assumes that the matching concepts appear in certain contexts within the matching ontologies. In other words, the usefulness of the method can be seen when there are concepts having identical names but different meanings depending on their place in the ontology (the context). For example, browsing through a music classification one can find two identically named music styles *Experimental*, whereas they are totally different because one is a style in the genre *Jazz*, and the other in the genre *Techno*, as illustrated on Figure 2.1. Despite the identical names these styles will be detected as different by

Ctx-match, because their parent concepts are semantically very different. Another advantage of the method can be seen in the case where two concepts have the same meaning but are named differently, because Ctx-match can still discover them to be equivalent. Ctx-match is presented in [Bouquet et al., 2003] together with promising experimental results. In the experiments Wordnet served as a background knowledge source, and the linguistic interpretation was done using a simple terminological matcher.

Ctx-match is similar with our approach in the sense that it follows a two-steps scheme: anchoring and relation deriving. The difference is that Ctx-match finds the matches by satisfiability testing within a logic language framework, whereas we do that (deriving relations) by reasoning over the relations in a background knowledge structure. Reasoning in general, besides the logic-based deduction, anticipates the use of domain specific relations like *part-whole*. Interpreting them is application specific, opposite to the logic-based relations which are domain and application independent. Positive side of Ctx-match is that it is generic, but the drawback is its requirement that the background knowledge structure and the resulting match follow the rigid logic framework.

Another tool based on Ctx-match, also developed at the University of Trento [Giunchiglia et al., 2004], is called S-Match. As explained in [Giunchiglia et al., 2004], S-Match is optimized re-implementation of Ctx-match with a few added functionalities.

- **Matching through a reference.** In [Zhang and Bodenreider, 2005] the authors elaborate on a paradigm where the interoperability between different ontologies is realized through matching to a reference ontology in the domain. Instead of matching each pair of ontologies to one another, they are all matched to a comprehensive domain reference ontology, yielding the mutual matches through this reference. If there are  $n$  ontologies in the domain, the number of pairs of ontologies to be matched is  $O(n^2)$ , while through a reference this number reduces to  $n$ . This looks nice in theory, but can it actually work in practice?

The answer given in [Zhang and Bodenreider, 2005] is yes, this scheme can be successfully applied in a practical setting. The hypothesis that indirect matching can serve as reliable alternative to the direct matching was tested in an experimental setup. Two medical ontologies were matched, using much larger and better covering ontology as a background knowledge. Two experiments were conducted: direct matching and indirect matching through the reference. The result of both ex-

periments was a set of pairs of equivalent concepts from the matching ontologies.

The first experiment proceeded in two phases. First, all the concept names were normalized and enriched with synonyms from the UMLS. Second, all the concept pairs from the matching ontologies were compared for equality of their normalized and enriched names. When this equality was detected, the pair of concepts was reported a match.

The second experiment proceeded in two phases as well. In the first phase the same direct matching procedure from the previous experiment was applied between each of the matching ontologies and the reference. In the second phase, an indirect match was deduced each time a source and target concepts were anchored (directly matched in the first phase) to the same concept in the reference.

The matches found in both experiments made up around 85% of all the matches found by at least one experiment. This supports the claim that it is feasible to reliably replace direct matching by an indirect through a reference, but of course we must have a reference domain ontology to do that.

- **Using the Semantic Web as background knowledge.** Inspired by our own work, in [Sabou et al., 2006] the authors explore the process of using large number of background ontologies in the matching. In the experimental setup they used the Semantic Web as a source of these ontologies. In particular, they were detected using the ontology search engine Swoogle. The focus of the work is on the feasibility of this detection. The concept names from the matching ontologies serve as search terms for finding background knowledge for the matching.

This study supports our hypothesis that the Semantic Web can be used as a source for the background knowledge ontologies.

- **Discovering missing background knowledge.** An algorithm to discover and use missing background knowledge fully automatically during the matching process is proposed in [Giunchiglia et al., 2006]. The discovery proceeds iteratively using a heuristic.

In each iteration potentially missing background knowledge is detected. A pair of matching concepts is considered a candidate match if its elements are not matched, and the majority of their subconcepts in the hierarchy below are matched. Putting it the other way around, matched concepts indicate that their parents should be matched as well. When the potentially missing knowledge is detected it is added

to the background knowledge, i.e. the candidate match is accepted as a match, and the matching process goes on iteratively. Once missing background knowledge is found it can be reused in the future.

This study indicates that missing background knowledge is an obstacle that not necessarily prevents the success of using background knowledge in the matching.

One addressed concern in the literature is the discovery of background knowledge. As shown above, it can be done either by finding candidate background ontologies or automatic discovery and use of missing background knowledge. In this thesis we will discuss the features that indicate the selection of a successful background knowledge, Chapter 8.

The Ctx-Match system investigates the possibility of putting the whole matching process in a strict logical framework. Complementary, in our work we primarily assume that the reasoning uses domain dependent knowledge and goes beyond the basic logic deduction. The domain dependent reasoning is always some sort of extension to the logic framework. Through experiments in a specialized domain in Chapter 6, we observe that domain-dependent knowledge can provide a substantial contribution to the matching performance.

Furthermore, in Chapter 5 this thesis presents an investigation in the use of multiple background sources for the use of two different cases: a specialized domain with selected background ontologies from the same domain, and a general domain with background ontologies coming from different but related domains. Although in the literature multiple background sources have been used, we are not aware of such a study that investigates the system behavior with respect to the use of multiple sources.

Except for the Ctx-match whose work was published earlier, the other work on matching through a background ontology is conducted and published during the same time period as the research reported in this thesis.

# Chapter 3

## Formal framework

The last chapter provided an overview on the existing ontology matching techniques, emphasizing the approach of using background knowledge. In this chapter we will focus further on this idea. First, we will describe its underlying intuition with an example, then we will provide a formal framework of how to do ontology matching using background knowledge in the form of an ontology, and finally we will introduce measures of quality of a matching result and discuss the ways to measure the contribution of the background knowledge in the matching process.

### 3.1 Using ontology as a background knowledge

In the general scheme of this approach we distinguish: *Source* (SRC), *Target* (TAR) and *Background Knowledge* (BK) ontologies, as illustrated on Figure 3.1. Source and target refer to the ontologies being matched. They are named differently, but the matching is usually symmetric and gives them no special treatment. Swapping their places would only invert the matching result. However, there are use-cases where source and target are distinguished<sup>1</sup>.

In the matching process, the source and target are first bonded to the background knowledge, and then the actual matches are found through this bond and the background knowledge structure. Hence, this process proceeds in two steps: *anchoring* - bonding the matching ontologies to the background

---

<sup>1</sup>For instance, when the matching is done with the purpose of reclassifying instances from the source to the target ontology, the source and target are distinguished and they do not have equal roles.



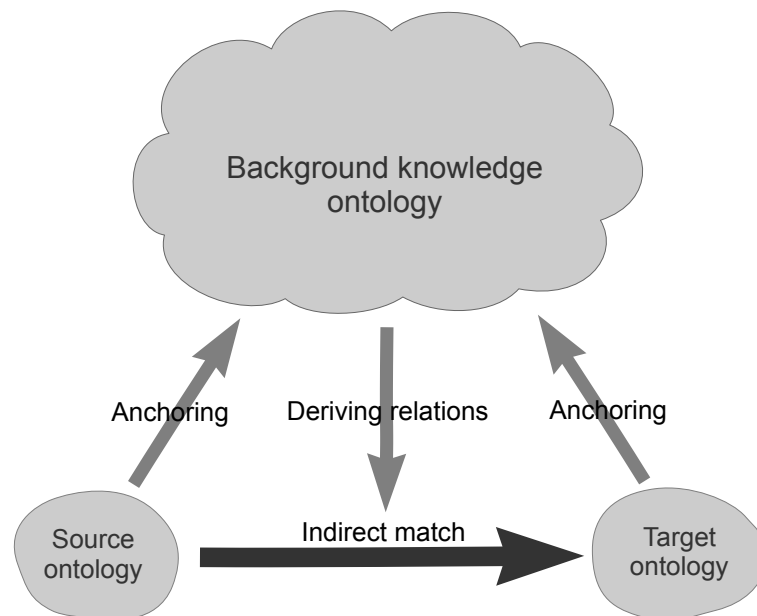


Figure 3.1: Scheme of ontology matching using background knowledge.

knowledge, and *deriving relations* - finding the actual matches between source and target through the background knowledge, see Figure 3.1.

**Anchoring** is the process of assigning appropriate background concepts to the matching (source and target) concepts. The assigned background concept, which we simply call anchor, might have the same meaning as the concept it is being assigned to, or describe one aspect of it. This depends on the structure of the background knowledge.

When the background knowledge domain allows to find anchors with equivalent meaning to the matching concepts, the anchoring can be performed using an existing ontology matching technique. Any of the techniques discussed in Chapter 2 is a potential candidate to perform this task.

When the domain of the background knowledge describes an aspect of the matching concepts, the anchoring will require a different matching approach. In the experiments reported in Chapter 4, partial lexical matching adjusted to the Dutch language was used for the task. The matching concepts were medical problems, and the background knowledge described causes of medical problems, among other things. The partial lexical matching was successfully used to assign causes (background concepts) to the medical problems

(matching concepts). For example, the cause BK: **Drugs** was attributed to the medical problem SRC: **Drugs overdose** in this way.

In general, the anchoring can be any matching that assigns background concepts to the matching concepts. This process is required to provide a sufficient bond between the matching ontologies and the background knowledge, which when combined with the background structure will enable the discovery of indirect matches, in particular those which are difficult or impossible to find using the classical direct matching techniques.

**Deriving relations** is the process of discovering the actual matches between the source and target concepts based on the combination of their anchors and the background structure. Related anchors through the background structure, indicate that their corresponding matching concepts are related as well, however, not any kind of relatedness between the anchors would imply a match. Discovering these relations between the anchors that are useful (i.e. imply a match) is a case-specific task that depends on what kind of relatedness can be discovered through the background knowledge, and the purpose of the matching. In some cases detecting very similar matching concepts is sufficient, like when integrating two web directory classifications for example, while in others very specific domain relations are required like *Genre - Typical instrument* in the music domain. Checking whether and how the anchors relate to one another is a reasoning task in the background knowledge, it exploits the background knowledge structure.

To make this process concrete we use a simple realistic example taken from the medical domain. It is illustrated on Figure 3.2. The source ontology contains a concept labeled SRC: **Brain** and the target ontology contains a concept labeled TAR: **Head**. An obvious match that one would be interested to find is that the concept SRC: **Brain** has a narrower meaning than the concept TAR: **Head**. In the source ontology there is no concept named SRC: **Head**, and in the target ontology there is a concept named TAR: **Brain** but it is not related to the target concept TAR: **Head** (instead it is related to TAR: **Central Nervous System**). The desired match cannot be discovered lexically (matching lexically would only find SRC: **Brain** *equivalent-to* TAR: **Brain**), and the structure of the matching ontologies is insufficient to indicate it. Given the situation, this match is impossible to find using automatic direct matching techniques. However, using background ontology it can be discovered. In the experiments the background knowledge was an ontology which comprehensively describes the domain of anatomy. Its structure gave clear indication of the match. The process, shown on Figure 3.2, goes as follows: the source concept SRC: **Brain** is anchored to a background concept BK: **Brain**, and a target concept TAR: **Head**

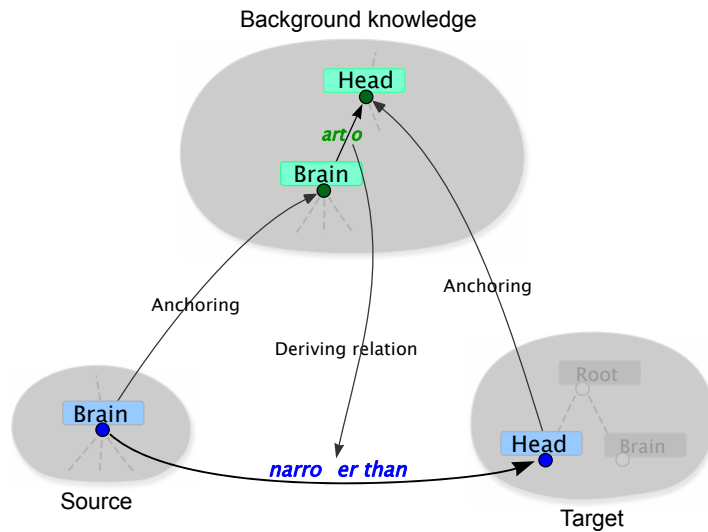


Figure 3.2: Scheme of ontology matching using background knowledge.

is anchored to a background concept BK: Head. The background knowledge reveals a relation between the two anchors BK: Brain *part-of* BK: Head, and since the relation *part-of* is a special kind of *narrower-than* relation, we conclude the desired match: SRC: Brain has a narrower meaning than TAR: Head. Using the background knowledge is crucial in this case to automatically discover the desired match.

Now, since the anchoring step consists of two matching problems (source and target to the background knowledge), it would seem that we are replacing one matching problem (source to target) by two. This would only makes sense if the subproblem to create a successful anchoring can be done easier than the original matching problem (source to target). A priori there is no reason to believe this, but in the empirical work in the later chapters we will show that indeed a simple matching techniques used for anchoring are good enough to subsequently create a good matching for the original problem.

As the example suggests, of particular interest in this approach is exploiting the structure of the background ontology. It takes place in the deriving relations step, when checking for relatedness between the anchors. Now, natural question is why would one use background knowledge at all? Does it contribute something to the existing matching techniques? The existing direct matching approaches fail if there is no sufficient lexical overlap between the matching concepts, and the other means - structure in the source and target, and the instance data are also insufficient to provide evidence for a

match. The use of background knowledge is a way to bridge this gap. Useful relations omitted in the direct matching can be found through a (rich) background knowledge. The background knowledge is assumed to hold richer structure than that of the matching ontologies compounded. Richer does not mean simply larger in size (although larger size is an indicator of the richness of an ontology), but more comprehensively describing the domain or part of it. It might be the case that background knowledge is smaller in size but richer in a special part of the domain. For example, when matching anatomy ontologies, a comprehensive ontology describing the *Vascular System* can be used as background knowledge. It would help in connecting the matching concepts describing the Vascular System (unlikely the others). Even though with smaller size than the matching ontologies, this background knowledge can potentially have a large contribution.

Next, we formally define the components in the matching framework. We will use these in the rest of the thesis.

## 3.2 Terminology of the framework

In our matching model we assume that the involved ontologies are structured according to the same principles. Each of them consists of a set of concepts interrelated with various types of relations, and the concepts and relations are described through text descriptions - labels.

Concepts are the building blocks of any model, or any domain description. A concept is an abstraction representing an object or a certain phenomenon. They introduce a perspective, a way to look at and communicate about the real world.

**Definition 1. Concept** is a class of things grouped together due to some shared property. It is named with a label, and sometimes with additional alternative names (synonyms). Besides the name(s), the meaning of a concept is determined by its semantic neighborhood, that is how it is asserted to relate to the other concepts in the ontology. We will refer to concepts in the following ways:

- Capital italic letters  $X$ ,  $Y$ ... when referring to an arbitrary concept
- $X^{\text{ONT}}$  when referring to a concept from a specific ontology, in this case ONT

- Using its label in Sans Serif font, like Temporal lobe, Alternative Rock, Person, House, etc.
- Using the label preceded with the ontology where the concept comes from, and delimited with a colon, like `ONT:Temporal lobe`, which refers to the concept Temporal lobe from the specific ontology ONT

**Definition 2. Relation instance** is a triple  $(X \text{ relation } Y)$ , where  $X$  and  $Y$  are concepts, and  $\text{relation} \in \mathcal{T}$  is a relation type.  $\mathcal{T}$  is the universal set of all relation types. The relation instance  $(X \text{ relation } Y)$  is interpreted as the concept  $X$  is related through the relation type  $\text{relation}$  to the concept  $Y$ .

When clear from the context we will call the relation instances simply relations. We will refer to a relation in the following ways:

- $(X \text{ relation } Y)$  when referring to a relation instance of the type  $\text{relation}$  between two arbitrary concepts.
- $(X \sim Y)$  when referring to a relation instance of a type which is denoted by the symbol  $\sim$ . A simple example is the relation type *subset-of* with the corresponding symbol " $\subseteq$ " :  $(X \subseteq Y)$ . In this case the corresponding concepts should denote sets.

Different relation types have different meaning, i.e. carry different semantics. Some are given special treatment, for example the relation types corresponding to hierarchical ordering are usually transitive. The mereological relation *part-of* is used in many domains to organize the content through spatial division. For example, the geographical concepts like countries, states, cities, regions, rivers, mountains, etc. are naturally organized through *part-of* relation type. Examples of relation types used in this thesis are:

- $(X \equiv Y)$  - the two concepts are equivalent, they have the same meaning,
- $(X \preceq Y)$  (with inverse:  $\succeq$ ), also written as  $(X \text{ is-narrower-than } Y)$  - the first has a narrower meaning than the second in semantic sense.

Various other relation types are used in the existing ontologies as well. The above mentioned types are independent of the domain (or cover very broad range of domains), but some are confined to certain domain. Domain dependent relations can be very specific, for instance *Genre-Artist* relation, where an artist is assigned a music genre.

Relations can connect concepts from one single, or two different ontologies. In the first case are relations that constitute the internal structure of

the ontologies. In the second case are relations that represent a matching between different ontologies. They can serve as a bridge between them.

As discussed in [Gruber, 1993], ontology is a formal specification of a domain conceptualization. It aims to provide means for effective communication within a domain. It is a controlled vocabulary that describes the entities of the domain and the relations between them in a formal way. The ontologies ask their users for commitment to use the vocabulary in a consistent way for knowledge sharing.

**Definition 3. Ontology** is a pair of sets:  $\text{ONT}(\mathcal{C}, \mathcal{R})$ .  $\mathcal{C}$  is a set of concepts,  $\mathcal{R}$  is the set of relations among these concepts.

We will refer to ontologies in the following ways:

- Using its full name, like Foundational Model of Anatomy (or the name in italic *Foundational Model of Anatomy*)
- Short form of the name in Sans Serif font, like ONT for an arbitrary ontology, or FMA for the particular ontology *Foundational Model of Anatomy*.
- $\text{ONT}(\mathcal{C}, \mathcal{R})$  when referring to an ontology with particular sets  $\mathcal{C}$  of concepts and  $\mathcal{R}$  of relations. When referring to  $\mathcal{C}$  or  $\mathcal{R}$  of a specific ontology, we will write them as  $\mathcal{C}^{\text{ONT}}$  and  $\mathcal{R}^{\text{ONT}}$ .

In the traditional view on ontology (such as in description logic), a special treatment is given to the subsumption relations that constitute the hierarchical structure of the ontology<sup>2</sup>.

**Definition 4. Ontology match** between two ontologies  $\mathbf{S}$  and  $\mathbf{T}$  is a set of relation instances:

$$M \subseteq \mathcal{C}^{\mathbf{S}} \times \mathcal{T} \times \mathcal{C}^{\mathbf{T}} \quad (3.1)$$

Each element in this set  $(X \ r \ Y) : X \in \mathcal{C}^{\mathbf{S}}, r \in \mathcal{T}, Y \in \mathcal{C}^{\mathbf{T}}$  we call a *match* between  $X$  and  $Y$ , or,  $X$  is matched to  $Y$ , through the relation type  $r$ . We will write it as  $X \xrightarrow{r} Y$ , or,  $X \rightarrow Y$  when the relation type of the match is known from the context. We will denote the ontology match between two ontologies  $\mathbf{S}$  and  $\mathbf{T}$  with  $\mathbf{S} \rightarrow \mathbf{T}$ .

An ontology match is the result of any ontology matching technique. In practice, it plays the role of a bridge between different ontologies. However, no standard definition exists for ontology match. In the literature more

---

<sup>2</sup><http://www.w3.org/TR/owl-features/>

restricted definitions can be encountered. For example, when assuming that at most one match can be established between a single pair of concepts, the ontology match can be defined as a function that assigns a relation type to each concept pair, [Shvaiko and Euzenat, 2005]. In other cases, the relation types used are restricted, and match is considered only when the concepts have equivalent meaning.

Two specific ontology matches are of particular interest to our approach. They correspond to the two phases of the matching - anchoring and deriving relations.

**Definition 5. Anchor-relation** is an ontology match from the source or target ontology to the background knowledge, respectively. We call them *source anchor-relations* and *target anchor-relations*:

$$A^{\text{SRC} \rightarrow \text{BK}} \subseteq \mathcal{C}^{\text{SRC}} \times \mathcal{T} \times \mathcal{C}^{\text{BK}} \quad (3.2)$$

$$A^{\text{TAR} \rightarrow \text{BK}} \subseteq \mathcal{C}^{\text{TAR}} \times \mathcal{T} \times \mathcal{C}^{\text{BK}} \quad (3.3)$$

To denote an anchor-relation we will use the connective  $\mapsto$ , so an anchoring relation between two arbitrary concepts  $X$  and  $Y$  will be written as  $(X \overset{\sim}{\mapsto} Y)$  with " $\sim$ " being the symbol of the relation type in this anchoring match. If known from the context the symbol will be omitted, and the relation will be simply written  $(X \mapsto Y)$ .

**Definition 6. Deriving relations** is an ontology match between the source and the target ontology.

$$D \subseteq \mathcal{C}^{\text{SRC}} \times \mathcal{T} \times \mathcal{C}^{\text{TAR}} \quad (3.4)$$

It establishes the actual indirect match between the matching ontologies. It is created by combining the anchor-relations and the background knowledge ontology. We call it an indirect match because it is not established in a direct comparison, but an indirect one through the background knowledge. Deriving relations is the driving engine in the approach of using background knowledge in ontology matching.

Deriving relations can be simple, like checking if the anchors of the matching concepts are the same. It can also be complex, like checking for relations among anchors where a single matching concept can have multiple anchors and the required matches can be obtained only by combined comparison of these anchors. A ranking scheme to estimate the matching strength can also be involved, see Chapter 4.

Once created, the problem is to evaluate the ontology match. Next, we discuss the quality measures of an ontology match.

### 3.3 Evaluation of ontology matching

Given two matching ontologies, source - SRC, and target - TAR, we observe the sets of matches which are of interest to the evaluation. The set of all possible matches between SRC and TAR is:

$$Candidates = \{(XrelY) \mid X \in \mathcal{C}^{SRC}, rel \in \mathcal{T}, Y \in \mathcal{C}^{TAR}\} \quad (3.5)$$

For a given purpose not all of these candidates are useful. With respect to an application, we divide this set in matches useful for the application - *desired matches*, and the others not useful for the application - *undesired matches*.

$$Desired \subseteq Candidates \quad (3.6)$$

$$Undesired \subseteq Candidates \quad (3.7)$$

These two sets form a partition of *Candidates*:

$$Desired \cap Undesired = \emptyset \quad (3.8)$$

$$Desired \cup Undesired = Candidates \quad (3.9)$$

The desired and undesired matches in the literature are named using different terminology: correct and incorrect, true and wrong, good and bad matches, etc. [Euzenat, 2007]. The set of desired matches is also called Gold Standard for the system performance. The result of ontology matching is a set of *found matches*:

$$Found \subseteq Candidates \quad (3.10)$$

These sets are illustrated in Figure 3.3. What characterizes the success of the matching is the correspondence between *Found* and *Desired* matches. The final goal is to find exactly the desired matches as a result  $Found = Desired$ . Therefore, a good matching would maximize the overlap between *Found* and *Desired*, while minimizing the overlap between *Found* and *Undesired*. To characterize the degree of success we adopt two notions from the information retrieval field: *precision* and *recall*.

#### 3.3.1 Qualities of matching: Precision and Recall

In Information Retrieval (IR) the precision and recall are measures on performance of document retrieval [Baeza-Yates et al., 1999]. They rely on a



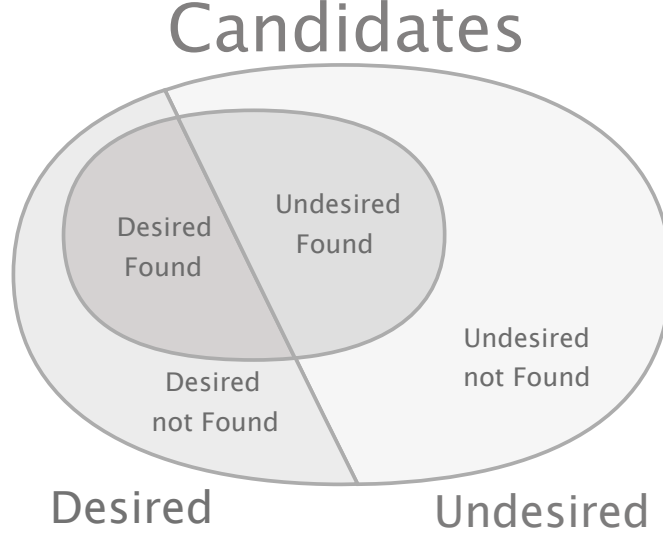


Figure 3.3: In ontology matching the matches can be desired or undesired, and independently, found or not found by the matching method.

collection of documents and a query for which the relevancy of the document is known, assuming binary relevancy: a document is either relevant or non-relevant. The precision is defined as:

$$\mathbf{Precision}^{(IR)} = \frac{|Retrieved\_documents \cap Relevant\_documents|}{|Retrieved\_documents|} \quad (3.11)$$

It is the proportion of relevant documents in the set of all retrieved documents retrieved. Its value is between 0 and 1 inclusive, where 0 means no relevant document has been retrieved, and 1 means only relevant documents have been retrieved. The recall is defined as follows:

$$\mathbf{Recall}^{(IR)} = \frac{|Retrieved\_documents \cap Relevant\_documents|}{|Relevant\_documents|} \quad (3.12)$$

It is the proportion of retrieved documents in the set of all relevant documents. As the precision, its value is also between 0 and 1 inclusive, where 0 means no relevant documents were retrieved, and 1 means all the relevant documents were retrieved.

As can be observed from the formula, the precision is not defined when there are no retrieved documents. In that case the measure does not have a meaning, namely, when there are no matches we cannot characterize their

quality (that would be the same as to characterize the elements in the empty set). Similarly, the recall is not defined and does not have a meaning when there are no relevant documents to be retrieved.

In the ontology matching we define these measures through the sets of matches previously discussed.

**Definition 7. Precision** is the proportion of desired and found matches, to all the found matches:

$$\mathbf{Precision} = \frac{|Desired \cap Found|}{|Found|} \quad (3.13)$$

**Definition 8. Recall** is the proportion of desired and found matches, to all the desired matches:

$$\mathbf{Recall} = \frac{|Desired \cap Found|}{|Desired|} \quad (3.14)$$

The interpretation of these measures is analogous to those in Information Retrieval: the matches correspond to the documents, and the desired matches correspond to the relevant documents. Often, the precision and recall are expressed in terms of percentage, ranging from 0% to 100%.

The precision and recall are not independent between each other. It is easy to get either of them at 100%: if the matching returns no results the precision is 100%, and when all the candidates are returned the recall is 100%. Often the matching systems can be tuned on a gradual scale between these two extremes, one to produce perfect precision and the other to produce perfect recall. When gradually turning from one to the other extreme, the plotted values of precision and recall provide means to visualize this trade-off in the so-called recall-precision diagrams.

Other measures, such as fallout, F-measure, noise, silence, etc. can be derived from precision and recall [Do et al., 2003]. Relaxed precision and recall variants are discussed in [Ehrig and Euzenat, 2005].

**Example of precision and recall** To explain the meaning of precision and recall we take a simple hypothetical example. The matching ontologies are parts from imagined registering systems in two hospitals:

- SRC: Hypothermia
- SRC: Treated person
- SRC: High body temperature
- SRC: Tumor

TAR: Hyperthermia  
 TAR: Patient  
 TAR: Tumor

We use a classical lexical matcher which is primarily based on edit distance measure (discussed in Section 2.1.1). It produces the following result:

SRC: Hypothermia  $\equiv$  TAR: Hyperthermia (*undesired*)  
 SRC: Tumor  $\equiv$  TAR: Tumor (*desired*)

The matcher detects that SRC:Hypothermia and TAR:Hyperthermia are lexically similar enough (edit distance of 2) to deduce that they have the same meaning. However, that is wrong because the two terms have different meanings, making the derived match undesired. The second match is straight forward, SRC:Tumor has the same label as TAR:Tumor (edit distance 0) deducing they match to one another. On the other hand, the desired result in this case is:

SRC: High body temperature  $\equiv$  TAR: Hyperthermia (*not found*)  
 SRC: Treated person  $\equiv$  TAR: Patient (*not found*)  
 SRC: Tumor  $\equiv$  TAR: Tumor (*found*)

The matcher found 1 desired, 1 undesired match, and did not find 2 desired matches. We calculate the values of precision and recall:

$$\mathbf{Precision} = \frac{1}{2} = 50\% \quad (3.15)$$

$$\mathbf{Recall} = \frac{1}{3} = 33.3\% \quad (3.16)$$

The precision shows that 50% of all the matches are correct, and the recall shows that 33.3% of all the correct matches were found.

### 3.3.2 Matching Sample Evaluation

In practice, it is often too costly to manually assess all the mappings. The most common way to overcome this problem is by sampling. A random sample from the whole set of mappings is drawn, and then manually assessed. The results are generalized to estimate the system performance on the entire set. Given a set of matches  $M$ , and a sample drawn from it  $M' \subseteq M$ , the calculated precision and recall on the sample  $M'$  are estimators of those of  $M$ :

$$\mathbf{Precision}(M) \approx \mathbf{Precision}(M')$$

$$\mathbf{Recall}(M) \approx \mathbf{Recall}(M')$$

The sample has to be representative of the entire set, because otherwise the estimation will be skewed. Choosing a completely random sample is called the *simple random sampling* method. A problem occurs when different matchings exhibit different importance with respect to the matching task. In this case a better sampling technique will be the *stratified random sampling*. It first divides the entire population of matches into groups (strata) that exhibit similar importance, and then draws a sample that represents each strata proportionally to its size. For extensive discussion on the different sampling techniques see [Cochran, 1977]. Besides the sampling method, the sample size is another key factor in the evaluation. Reducing the evaluation size reduces the effort, but also reduces the significance of the evaluation. It is a trade-off between the sample size and the corresponding evaluation reliability. Statistics provide mechanism to quantify this trade-off, see [Van Rijsbergen, 1979, Fleiss et al., 1981].

Depending on the measure being estimated, different evaluation samples are drawn, as the formulas to compute these measures require. To estimate the precision, a sample from the set of *Found* matches is drawn and (manually) evaluated. For recall, a sample set of the *Desired* matches (i.e. the Gold Standard) is constructed, and the matching is challenged against it.

### 3.4 Added value of background knowledge

In order to justify its use, the approach of using background knowledge has to provide an added value to the existing matching techniques. Before investigating how much help it can provide and under which circumstances, we will first discuss what we consider as help, i.e., when, and how much is one matching better than another. Note that this is a general problem not specific to the approach, how can we compare two matchings against one another?

**Change in precision and recall** One way to compare two ontology matchings  $M_1$  and  $M_2$  is by observing the change in precision and recall. As discussed earlier, the ideal case exhibits 100% precision and 100% recall. Then better matching simply means closer to the ideal case:  $M_1$  is better than  $M_2$  if:

$$\mathbf{Precision}(M_1) > \mathbf{Precision}(M_2)$$

$$\mathbf{Recall}(M_1) > \mathbf{Recall}(M_2)$$

The difficult case is when one quality improves while the other drops. Then there is no straight forward way to judge which one is better, if any. In this case the improvement depends entirely on the application. A pair of values for precision and recall corresponds to a point in the precision-recall plane, and the set of all such pairs equally appreciated for the application can be visualized as a curve in the precision-recall plane. The different curves do not cross, and have a larger appreciation with increasing distance from the (0%, 0%) point in the plane.

The recent results of OAEI<sup>3</sup> show that the precision and recall performance of the state-of-the-art matching techniques depends on the matching task. When richer relations were asked in the matching, namely matching the concepts with *broader-than* and *narrower-than* relations, the techniques exhibited low recall<sup>4</sup>. For this reason, currently, increasing the recall can be seen as a key challenge.

**Baseline matching** Improvement of performance can only be observed in comparison with a baseline. In this thesis we consider two candidate baselines as the most prominent to challenge and analyze the approach of using background knowledge: state-of-the-art matching tools, and simple direct ontology matching. Each of them for a different reason.

Boosting the performance of state-of-the-art matching techniques is the goal of each new technique introduced to the field. There is a difficulty when comparing with state-of-the-art tools, namely, their results are difficult to interpret because they usually employ multiple matching strategies to produce the final result<sup>5</sup>. When using a transparent technique such as a simple lexical matching as a baseline, better analysis can be conducted on the improvement as each base-line match can be explained. Yet, that does not prevent the use of any matching technique as a baseline. In some cases even an indirect matching can be used as a baseline, which would then turn the comparison into a measurement of which background knowledge produces a better matching.

---

<sup>3</sup><http://oaei.ontologymatching.org/2006/>

<sup>4</sup>In 2006 the recall of all the participating tools was not above 50%

<sup>5</sup>For this reason they are often referred to as a "black box"

# Chapter 4

## Comparison with state-of-the-art matchers

In the previous chapter we introduced a formal framework for ontology matching by using background knowledge in the form of an ontology. In this chapter we will use this framework to show that using background knowledge can produce a better matching than the state-of-the-art matching tools on a representative realistic case study. The results from OAEI<sup>1</sup> in 2006 [Euzenat et al., 2006] showed that the current tools still have difficulties when challenged with realistic matching tasks. On our case study the state-of-the-art tools indeed fail to produce successful matching, while an approach that uses background knowledge succeeds. We will discuss what causes the state-of-the-art tools to fail, and how does the use of background knowledge overcome these limitations. This chapter is based on the work published in [Aleksovski et al., 2006a].

### 4.1 Introduction

We already discussed in the Chapter 2 that the majority of the current matching tools use a combination of terminological and structural methods, where the lexical overlap is used to produce an initial matching, which is subsequently improved by using the structure of the ontologies to be matched. Hence, these tools crucially rely on two assumptions: *(i)* sufficient lexical overlap exists between the source and target ontology, and *(ii)* source and target ontology have sufficient structure. We will present a case study where

---

<sup>1</sup>Ontology Alignment Evaluation Initiative <http://oei.ontologymatching.org/>

neither of these assumptions hold. In this case study, not only there is insufficient lexical overlap between source and target, but more crucially, the matching ontologies contain no structure at all: they are simply lists of concepts, instead of richly structured ontologies. Consequently, the current state-of-the-art matching tools are expected to fail.

We believe that our case study is representative of many realistic cases. Experience with the Semantic Web applications shows that many of them rely on rather lightweight semantic structures, providing at most a hierarchy of terms, where often this hierarchy is only a 2-3 levels deep [Schlobach, 2004, Wang et al., 2006]. Hence, the reliance of existing matching tools on such structure is indeed an important limiting factor.

After showing the failure of a number of state-of-the-art matchers in our case study, we will present an algorithm for ontology matching based on the framework described in the previous chapter, that uses background knowledge in the form of ontology to compensate for the lack of structure found in the source and target vocabularies, as well as the freedom in choice of terminology. As explained in Section 3.1, the idea of this algorithm is to first lexically anchor the concepts from the source and target ontologies to the background knowledge (*anchoring phase*), then use the structure of this background knowledge to induce semantic relationships between the source and target concepts, and finally use these relationships to derive a matching between them (*deriving relations phase*). On our case study the experimental results of applying this algorithm show that it succeeds in producing sufficient matching, therefore improving the performance of the most recent matching tools. Exhibiting a better quality than the state-of-the-art tools makes it a promising approach to the ontology matching problem.

The rest of this chapter is organized as follows: in Section 4.2 we will describe our case study, in Section 4.3 we will describe the two base-line experiments: matching the source and target using simple lexical technique and using state-of-the-art matching tools, then in Section 4.4 we will describe our approach and compare its experimental results with the baseline, to finally conclude the chapter with Section 4.5.

## 4.2 Case study description

This case study posed the challenge to match two vocabularies: OLVG and AMC taken from the medical domain. A rich and highly structured ontology DICE was used as a background knowledge in our algorithm.

The matching vocabularies are lists of reasons for admission to the Intensive Care Unit (ICU) of the two Amsterdam hospitals *OLVG*<sup>2</sup> and *AMC*<sup>3</sup>. A reason for admission describes a problem, why a patient was brought into the ICU. The situation of each patient arriving at the ICU in either hospital is described by assigning one or more problem descriptions from the list. The vocabularies together with the patient data are used for monitoring patient progress, for planning of required ICU resources, and for off-line nationwide quality comparison of different ICU's.

As a use-case the produced matching would enable to reclassify patients registered in the *OLVG* system, into the *AMC* system. The *AMC* system currently gets extended and further developed, and there is an initiative to make it widely used in more ICU's in the Netherlands. Automatic matching of other ICU vocabularies to the *AMC* is needed because the other systems will coexist with the *AMC* for some period of time, requiring such matching to be performed on a periodic basis. However, the automatically created matching will still be checked manually, and will be approved or disapproved by the experts. This automatic matching is needed to produce a suggestion that would reduce the workload given to the experts. For the purpose of our study, we state the matching problem as follows: For each *OLVG* concept, determine corresponding *AMC* concept which a physician would use to describe the same problem.

**Source vocabulary:** *OLVG*. The source vocabulary was the set of reasons for admission used in the Intensive Care Unit of the *OLVG* hospital, [Bosman et al., 1998]. It is a flat list with no structure consisting of concepts named with one label each. The list is partly based on the ICD-9-cm<sup>4</sup> vocabulary, and on the Dutch "Classificatie van Medisch Specialistische Verrichtingen" (CMSV)<sup>5</sup>, a classification of medical procedures. During its use in the past three years, the *OLVG* list has been extended with additional descriptions of medical conditions of patients at the ICU. The resulting list is a mixture of problem descriptions at several levels of abstraction with minor redundancy. It does not only contain reasons for admission to the ICU, but also other medical conditions that are relevant during the stay of patients at the ICU. We limited our experiments to the list-elements actually used at admission, identified as those terms that are used for describing patient's states during the first 24 hour of their stay. The resulting list contained 1399 problem

---

<sup>2</sup><http://www.olvg.nl/>

<sup>3</sup><http://www.amc.nl/>

<sup>4</sup><http://www.cdc.gov/nchs/about/otheract/icd9/abtcd9.htm>

<sup>5</sup>[http://www.nictiz.nl/kr\\_nictiz/2527](http://www.nictiz.nl/kr_nictiz/2527)



descriptions consisting of maximal 7 words each. 95% of these descriptions used no more than three words. The list is mainly in Dutch but also contains English terms.

**Target vocabulary:** AMC. The target vocabulary was the set of reasons for admission used in the Intensive Care Unit in the AMC hospital. Similarly with the source vocabulary, it consists of a flat list of 1460 reasons for admission. The list was codeveloped with the DICE ontology. This codevelopment included a linking of the vocabulary into the ontology. In our experiments we used this linking as an additional anchoring schema.

**Background ontology:** DICE. As a background knowledge we used part of the DICE ontology, [de Keizer et al., 1999]. DICE has been developed by the Medical Informatics group at the AMC hospital. It is a medical terminology, formalized in OWL DL <sup>6</sup>, of roughly 2300 concepts, described by 5000 lexical terms. The concepts are related among each other with 4300 relational links of 50 different relation types. DICE mainly aims to cover concepts in the Intensive Care domain, and its core structure consists of five different hierarchies (called *aspects* in DICE): *abnormalities* (255 concepts), *medical procedures* (55 concepts), *anatomical locations* (1512 concepts), *body subsystems* (13 concepts), and *causes* (85 concepts). Note that the size of these hierarchies is 1920 concepts, while the size of DICE is 2300 because it also contains medical concepts with meaning outside of the intensive care. Each aspect hierarchy provides a domain of possible values for the aspect. These hierarchies are organized as tree structured taxonomies. If a concept is named with multiple terms, one of the terms functions as ‘preferred’ term - label, and the others as synonyms.

**Evaluation:** For measuring the performance of the various methods and tools, we created a Gold Standard solution for this problem. A medical expert was invited to create manually matches between the OLVG and AMC vocabularies for a selected subset of the OLVG list. Because the intended use case was to classify patients that have been registered with OLVG concepts in the AMC taxonomy, the expert was asked to specify for each OLVG concept the AMC concept that he would use to describe the medical problem.

The expert was given a random sample set of 200 concepts from the OLVG list. This set was created as follows: 30 concepts were selected because they

---

<sup>6</sup><http://www.w3.org/TR/owl-guide/>

were general medical problems, to which were added the top 15 most frequently used concepts from the OLVG vocabulary, and finally supplemented with randomly drawn 155 concepts to a total of 200 concepts. For these 200 concepts, the expert created matches for 125 concepts, leaving the other 75 “unknown”. For each of the matched OLVG concepts he proposed one AMC concept as the most appropriate match. No statements about alternates were made.

In the process of analyzing the data and conducting preliminary experiments we discovered correct matches for some of the OLVG concepts for which the expert did not suggest a match in the AMC list, and we also found matches which seemed like a refinement of those proposed by the expert. After ensuring the validity of these newly obtained matches using Internet resources, primarily Google<sup>7</sup> and Wikipedia<sup>8</sup>, we updated the Gold Standard accordingly<sup>9</sup>. As a result we ended up with a final set of 148 matches based on the set of 200 OLVG concepts. This yielded a Gold Standard on the order of about 10% of the entire OLVG vocabulary.

## 4.3 Baseline experiments

We performed two baseline experiments. In the first, we matched OLVG directly to AMC directly using a simple lexical technique based on a word matching. It was modified to face the challenges posed by the Dutch language. In the second experiment we matched OLVG to AMC using two state-of-the-art ontology matching tools. The experiments produced sets of OLVG  $\rightarrow$  AMC matches. We will now describe these experiments with the obtained results.

### 4.3.1 Experiment 1: Lexical matching

In this experiment we checked for a lexical correspondence between each pair of concepts from the matching vocabularies. We used a heuristic that compares the concept’s labels and synonyms. As a result it finds equivalent and partial lexical matches, defined below, between the comparing concepts.

---

<sup>7</sup><http://google.com/>

<sup>8</sup><http://wikipedia.org/>

<sup>9</sup>The fact that the expert did not find the best match for all the concepts from the sample set indicated that the manual matching task was difficult.

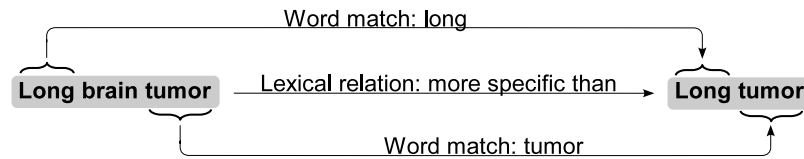


Figure 4.1: An example of matching two concepts lexically, by comparing the words in their labels: **Long brain tumor**  $\rightarrow$  **Long tumor**.

Before the actual comparison, the data were preprocessed by normalizing the concept's labels and synonyms. This normalization included: case standardization by converting all the alphabetic characters to their lower case counterpart, conversion of plural forms to singular (eliminating *s* or *en* at the end of the plural form) and stop-words elimination by discarding short words like *en*, *de* and *van*<sup>10</sup>.

Two arbitrary concepts  $X$  and  $Y$  were compared by comparing the words in their labels. If all the words in a label or synonym of  $X$  were found in a label or synonym of  $Y$ , a partial match was deduced between  $X$  and  $Y$  ( $X$  is concluded more general than  $Y$ ). Intuitively, the heuristic assumes that adding new words to an existing label only constrains the meaning. An example of a partial lexical match is shown in Figure 4.1. The first label consists of a superset of the words from the second, implying that the second have more general meaning than the first. Furthermore, the matching used some simple Dutch morphological rules to deal with the common Germanic construction of compound words that do not have a delimiting space between the words. This way, for example, **Hersentumor** (brain tumor) is partially matched to **Tumor**. When the labels of  $X$  and  $Y$  consisted of the same set of words, they were concluded lexically equivalent. The equivalence matches were favored over the partial matches.

In testing for equality of labels, we also allowed for edit-distance of two characters using Levenshtein string distance described in Section 2.1.1. This compensated for the typing mistakes in the lists.

Clearly, the superset of words heuristic is very simple, and it is not difficult to think of cases where it fails: In Figure 4.1, for example, **Long brain tumor** would also end up being a special case of **Brain**. However, in the context of the experiment it did not end up finding correspondences between such incompatible concepts.

<sup>10</sup>Note that the discarded stop-words and the plural forms are in the context of the Dutch language.

```

    Initially, the set of lexical matches is empty
1  lmatches :=  $\emptyset$ 

    Matching: find full and partial lexical matches
2  for every source concept  $X \in \mathcal{C}^{\text{OLVG}}$  do
3    for every target concept  $Y \in \mathcal{C}^{\text{AMC}}$ 
4      if FULLLEXMATCH( $X, Y$ ) then
5        lmatches  $\leftarrow (X \rightarrow Y)$ 
6      if full match was found then
7        continue for on line 2 (next OLVG concept)
8      for every target concept  $Y \in \mathcal{C}^{\text{AMC}}$  do
9        if PARTIALLEXMATCH( $X, Y$ ) then
10         lmatches  $\leftarrow (X \rightarrow Y)$ 
11  end for

```

Figure 4.2: MATCHDIRECTLY algorithm for matching OLVG to AMC vocabulary lexically.

The algorithm for lexically matching OLVG to AMC is shown in Figure 4.2. The two routines FULLLEXMATCH and PARTIALLEXMATCH compare two concepts as described above. Applied on our dataset it produced in total 582 matches  $\text{OLVG} \rightarrow \text{AMC}$ . Of these, 274 were matches of lexically equivalent concepts, and the remaining 308 were partial lexical matches.

### 4.3.2 Experiment 2: Matching with state-of-the-art tools

As discussed in Chapters 1 and 2, there is a variety of state-of-the-art matching tools currently available. We summarize two of the three most prominent tools that we considered in this study.

- FOAM is an ontology alignment framework to align fully or semi-automatically two or more OWL ontologies, developed by the university of Karlsruhe [Ehrig and Sure, 2005]. It is based on heuristics (similarity) of the individual entities (concepts, relations, and instances). As result, it returns pairs of aligned entities. It can handle ontologies within the DLP-fragment of OWL. Part of FOAM is a machine learning component that optionally takes user feedback into account.
- Falcon-AO [Jian et al., 2005] is an automatic ontology matching tool, developed by the South East University of China. It outperformed

all other ontology matchers in the 2005 and 2006 Ontology Alignment Evaluation Initiative [Euzenat et al., 2005, Euzenat et al., 2006]. Falcon-AO regards ontologies as graph-like structures, and then produces matches between elements in the two graphs that correspond to each other. Both of linguistic and structural similarity are taken into account. There are two matchers integrated in Falcon-AO: LMO for the linguistic similarity which is primarily based on edit distance, and GMO for the structural similarity, which performs a graph-based comparison.

The third tool we considered is S-Match which we described earlier in Section 2.2.2. As illustrated in the descriptions, these matching tools exploit a combination of syntactical and structural comparison techniques. However, when the sources to be matched are merely unstructured lists, the mixed approaches reduce to lexical comparison of labels. From this we derive the following:

*Hypothesis:* We expect that the results of the simple lexical matching are comparable to the performance of the most recent matching tools on weakly structured data, such as discussed in the beginning of this Chapter, since on these data the existing tools are also reduced to performing lexical matching only.

Only one of the tools, i.e. S-Match, exploits background knowledge to do the matching. However, in the current version S-Match can only use a predefined set of background knowledge sources, such as Wordnet and UMLS.

We have applied two of the ontology matching tools described above to the data in our case study. We loaded an OWL representation of both the OLVG list and the AMC list plus the DICE background knowledge into the tools.

When using FOAM to align the two lists, we initially got 326 matches, but those included symmetric matches, i.e. for each match  $X \rightarrow Y$  its symmetric equivalent  $Y \rightarrow X$  was also in the result set. Effectively FOAM found 159 matches. An analysis revealed that many obvious matches were missing because synonym labels were not recognized. To solve this, we regenerated the AMC list with separate concept definitions for each known synonym. This resulted in 696 matches.

When using Falcon-AO the experiments were performed in collaboration with Dr. Wei Hu, South East University of China. Due to memory constraints set by the tool, the experiment had to be split in subexperiments, each matching the OLVG ontology to a subset of the AMC ontology. The

experiment resulted in less than 100 matches. However, because all matches were necessary based on lexical measures only, we tried to use the lexical matcher component (LMO) in a stand-alone configuration. This has two advantages. First, the stand-alone LMO matcher is much more efficient, so that matching can run on the complete AMC ontology. Second, it returns a ranking of all matches found, and not just the ones that are above the threshold. As a result, we get many more matches, but many of them with a low confidence level. When using the stand-alone LMO matcher, 683 matches were returned.

Unfortunately, because S-Match is not freely available, we were not able to use S-Match on our dataset with the specific medical thesaurus as background knowledge.

	Number of matches OLVG $\rightarrow$ AMC
Lexical matching	582
FOAM	696
Falcon-AO	683

Figure 4.3: Amount of matches obtained in the baseline experiments

### 4.3.3 Discussion

The table in Figure 4.3 summarizes the amount of matches obtained in the baseline experiments. The numbers show that the large majority of matches obtained by the FOAM and Falcon-AO tools were already also found by our simple lexical matching, as they were also reduced to lexical matching on the test data.

**Evaluation against the Gold Standard** We evaluated the quality of the matches obtained in the baseline experiments by calculating their precision and recall quality measures as compared to the Gold Standard set of 148 matches. For each experiment, after selecting the subset of matches that corresponds to the Gold Standard (matches with the OLVG concept in the set of 200 concepts selected to construct the Gold Standard), we calculated the measures as described in Chapter 3:

- The lexical matching resulted in 70 found matches, 65 correct, and 5 incorrect. Following the formulas in 3.3.1, we calculated the measures:

$$\mathbf{Recall} = \frac{|Correct|}{|Desired|} = \frac{65}{148} = 44\% \quad (4.1)$$

$$\mathbf{Precision} = \frac{|Correct|}{|Found|} = \frac{65}{70} = 93\% \quad (4.2)$$

- FOAM resulted in 88 found matches, 41 correct, and 47 incorrect, therefore obtained **Recall** = 28% and **Precision** = 47%
- Falcon-AO resulted in 106 matches, 28 correct, and 78 incorrect, therefore obtained **Recall** = 19% and **Precision** = 26%.

The above figures are summarized in the table in Figure 4.4. Note that these figures cannot be used to judge the quality of FOAM and Falcon-AO, as we use them here in a scenario for which they were not designed, i.e. matching unstructured vocabularies.

	<b>Recall</b>	<b>Precision</b>
lexical matching	44%	93%
FOAM	28%	47%
Falcon-AO	19%	26%

Figure 4.4: Precision and recall measures characterizing the results obtained in the baseline matching experiments

Figure 4.4 shows that the recall of our lexical method is competitive with both FOAM and Falcon-AO. In particular, the LMO module of Falcon-AO retrieves much fewer of correct matches. The most likely explanation of this effect is that Falcon-AO limits itself to 1-1 matches, and shows a tendency to maximize the number of matched OLVG concepts, sometimes sacrificing a direct match. This is illustrated by the following example of matches produced by Falcon-AO on our case-study data:

OLVG: Oesofagus\_perforatie → AMC: Oesofagus ruptuur  
 OLVG: Oesophagus\_resectie → AMC: Oesofagus perforatie

although both by themselves reasonable, together these matches prevent the obvious match

OLVG: Oesofagus\_perforatie → AMC: Oesofagus perforatie

because Falcon limits itself to 1-1 matches: both of these concepts are already part of another match. The obvious solution would be to allow concepts to participate in multiple matches, producing an n-m matching as done by our lexical method.

## 4.4 Experiment 3: Matching using background knowledge

When matching the vocabularies indirectly, we followed the general scheme depicted on Figure 3.1 in Chapter 3. We used the five aspect taxonomies in DICE as background knowledge. In the anchoring step, we lexically matched the concepts from the OLVG and AMC vocabularies to the concepts each of the five DICE aspects taxonomies. For the AMC  $\mapsto$  DICE anchoring, besides the lexical anchoring, we used an additional anchoring schema that was created manually by experts from the AMC hospital. Important to notice is that the lexical matching procedure that we used is not crucial in the anchoring step, instead any matcher can be used to establish the anchoring matches. We used such simple lexical technique because our manual analysis indicated that using other more advanced techniques will only marginally change the result. After obtaining the anchors, we used the relationships specified in the DICE taxonomies to find indirect matches between the source and target concepts. Figure 4.5 depicts the scheme of this experiment, which is essentially the general scheme from Figure 3.1 instantiated for this particular case.

*Hypothesis:* We expect that using background knowledge improves the results obtained by applying direct lexical matching between the two vocabularies.

Next, we will describe how we performed the anchoring, and the deriving indirect matches in the experiments.

### 4.4.1 Anchoring

As explained, the concepts from the source and target vocabularies are anchored to the concepts in the background knowledge ontology. The automatic anchoring in our approach, is performed by the same lexical technique that we used in the first experiment in Section 4.3.1, which discovers equivalent and partial lexical matches between concepts.

When anchoring the OLVG vocabulary to DICE, we used the lexical technique solely. We found in total 549 of OLVG concepts anchored to DICE



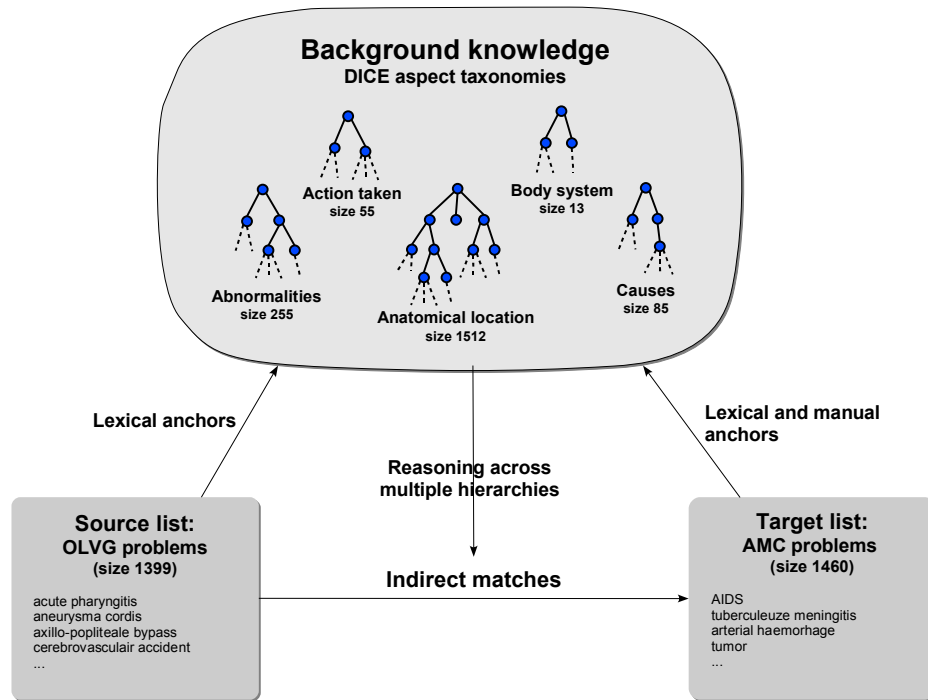


Figure 4.5: Matching the OLVG and AMC vocabularies using the five aspect taxonomies of DICE as background knowledge ontology

concepts, via 1298 anchors. For anchoring the AMC vocabulary to DICE we used a combined manual (expert-created) and automatic approach. An expert manually established 4568 DICE-anchors from the AMC vocabulary. We enhanced these anchor matches using the lexical matching technique and found a further 1248 new anchors, which increased the amount of anchors to a total of 5816, anchoring a total of 1404 AMC concepts.

Notice that the anchorings are many-to-many relations: a single concept from source or target vocabulary can have multiple anchor concepts in DICE, either in a single or in different aspect taxonomies. Table 4.6 shows how many concepts were anchored, and how often our lexical heuristics were able to establish anchorings to multiple DICE aspect taxonomies. Such anchoring in multiple aspects is important, because it will enable in the deriving relations step to use multiple DICE taxonomies to infer potential indirect matchings.

The table on Figure 4.6 shows that our simple lexical heuristics succeeded in constructing anchors for 39% of the OLVG vocabulary. This indicates that indeed our weak lexical heuristics are able to establish anchorings, to be used

#### 4.4. EXPERIMENT 3: MATCHING USING BACKGROUND KNOWLEDGE<sup>65</sup>

in the second step of our approach. The high percentage of anchoring for the AMC vocabulary, 96%, is due to the contribution of the manually constructed anchors.

	OLVG concepts	AMC concepts
anchored on 5 DICE aspects	0	2
anchored on 4 DICE aspects	0	198
anchored on 3 DICE aspects	4	711
anchored on 2 DICE aspects	144	285
anchored on 1 DICE aspect	401	208
total nr. of anchored concepts	549 (=39%)	1404 (=96%)
total nr. of anchoring relations	1298	5816

Figure 4.6: Results of the anchoring step

Figure 4.7 details how the anchors are distributed over the five DICE aspects (separate taxonomies). It shows that the anchors are very unevenly distributed over the various aspects (with only three anchors established from the OLVG vocabulary to aspect hierarchy on body-systems), and a similarly uneven relative contribution between expert-created and lexically found anchors across the different aspects (with again the body-systems aspect producing very few lexical but many expert-created anchors for the AMC vocabulary).

Aspect	AMC $\mapsto$ DICE			OLVG $\mapsto$ DICE
	Expert-man.	Additional lexical	Total	Lexical
<i>Abnormality</i>	1168	271	1439	354
<i>Action taken</i>	292	122	414	109
<i>Body system</i>	1217	2	1219	3
<i>Location</i>	1336	721	2057	772
<i>Cause</i>	555	132	687	60
	4568	1248	5816	1298

Figure 4.7: Distribution of anchors over the different DICE aspect taxonomies

The anchoring process adds structure in an implicit manner to the unstructured source and target vocabularies, and has established indirect relations between source and target concepts, through the relations between their anchors in the background knowledge ontology. We exploit this structure in the next step, to discover indirect matches between the source and target vocabulary concepts.

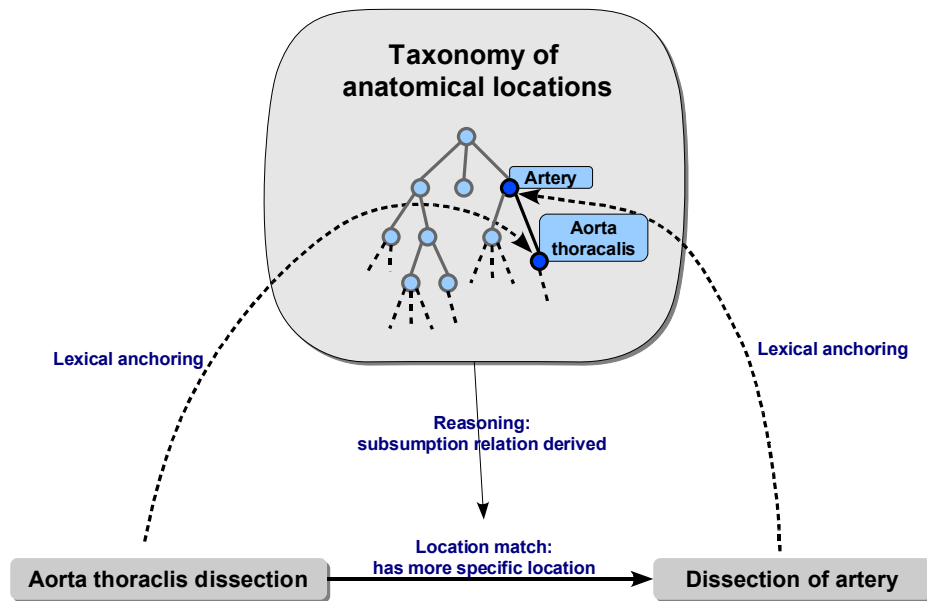


Figure 4.8: An example of relating two medical concepts using background knowledge. An indirect match is discovered using a taxonomy of anatomical locations.

#### 4.4.2 Deriving relations

In this phase we make use of the structure of the background knowledge DICE to discover the actual indirect matches between OLVG and AMC concepts. Simply, when comparing two concepts  $X$  and  $Y$  being anchored to  $X_1$  and  $Y_1$  respectively, we compare  $X_1$  and  $Y_1$ , and, if they are related, infer that  $X$  and  $Y$  are related as well.

An example is given in Figure 4.8: the concept Dissection of artery is found to have location Artery, and the concept Aorta thoracalis dissection is found to have location Aorta thoracalis. A relation is inferred between these two medical concepts, since they describe related anatomical locations: according to the background ontology, Aorta thoracalis is a kind of Artery. Hence, the source concept can be inferred to have a more specific location than the target concept. The use of background knowledge was essential to derive this match.

In general, both the source concept  $X$  and the target concept  $Y$  can be anchored to multiple anchors and the background knowledge could reveal relationships on anchors that represent different properties. On the one hand,

#### 4.4. EXPERIMENT 3: MATCHING USING BACKGROUND KNOWLEDGE<sup>67</sup>

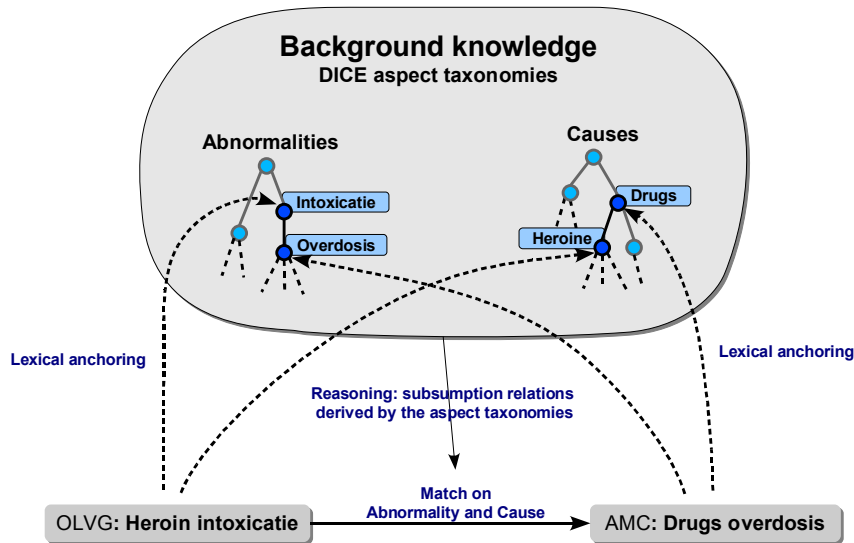


Figure 4.9: An example of matching two lexically unrelated concepts using background knowledge. The concepts are not equivalent but do have a close semantic relationship.

this makes the comparison more complex. If multiple anchors are related in similar ways, they reinforce that the main concepts  $X$  and  $Y$  are related in the same way. In case the multiple anchors are related in incompatible ways (e.g.  $X_1 \sqsubseteq Y_1$ , but  $X_2 \sqsupseteq Y_2$ ), a subsumption relation between the concepts cannot be inferred. However, they do reveal the source and target concepts have a strong relationship and are within some semantic distance.

This is illustrated in the example depicted in Figure 4.9, where we try to match the concepts OLVG: Heroin intoxicatie and AMC: Drugs overdosis. According to the background knowledge DICE, Heroin is a kind of Drugs, while Overdosis is a kind of Intoxicatie, i.e. the two aspects have a subsumption relationship to each other, but, in the reverse direction. Hence, the concepts are neither equivalent nor one subsuming the other. However, in the everyday understanding the two concepts do have a close semantic relationship. Note that these concepts do not have any lexical similarity - their describing labels consist of entirely disjoint sets of words, and even more, these words are not synonymous to each other but only have related meaning. It was only possible to discover this match by using the background knowledge.

The successful application of the method depends on the richness of the background knowledge. With increasing richness, the more likely it becomes that anchors can be established in the first place, but, more importantly,

```

    Initially, the set of indirect matches is empty
1  imatches :=  $\emptyset$ 

    Anchor OLVG and AMC to DICE
2   $\text{anch}^{\text{O} \rightarrow \text{D}}$  := MATCHDIRECTLY(OLVG, DICE)
3   $\text{anch}^{\text{A} \rightarrow \text{D}}$  := MATCHDIRECTLY(AMC, DICE)  $\cup$  expertMatches

    Indirect matching: find matches through the anchors in DICE
4  for every concept pair  $X \in \mathcal{C}^{\text{OLVG}}, Y \in \mathcal{C}^{\text{AMC}}$  do
5      if two anchors exist  $(X \mapsto X_1) \in \text{anch}^{\text{O} \rightarrow \text{D}}$  and  $(Y \mapsto Y_1) \in \text{anch}^{\text{A} \rightarrow \text{D}}$ 
6          such that  $X_1 \sqsubseteq Y_1$  or  $Y_1 \sqsubseteq X_1$  then
7          imatches  $\leftarrow (X \rightarrow Y)$ 

```

Figure 4.10: Algorithm for indirectly matching OLVG to AMC through DICE ontology as a background knowledge. In the anchoring phase it uses the method MATCHDIRECTLY described in 4.2

the more likely it becomes that an indirect match between the source and target can be found. Especially interesting are the indirect matches which are not discovered directly, because, as discussed in Section 3, they represent the contribution of the indirect matching. For this reason in our experiments when assessing the results of the indirect matching we included the directly obtained matches. This way we were able to assess the improvement gained by the use of the background knowledge.

Experience in practice has shown that concepts from two matching ontologies are rarely precisely equivalent, but rather have some (otherwise unspecified) semantic overlap. Consequently, finding such semantic relationships seems more useful for integration purposes, than finding precise equivalences.

The algorithm for finding indirect matches is shown in Figure 4.10. When applying this algorithm, as a result of the indirect matching between OLVG and AMC, a matching AMC concept was derived for 548 OLVG concepts. Of these, 413 matches were based on inference in a single DICE aspect, while 135 matches were supported by inference in two aspects (i.e. the inference in two DICE taxonomies produced support for the same match).

**Evaluation against the Gold Standard** The evaluation of the indirect matching against the Gold Standard is made more complicated by the fact that the n-m anchors can also produce n-m matches between source and target vocabularies. When a single OLVG concept matches with multiple concepts in the AMC vocabulary, we ranked the matchings as follows:

1. If the match corresponds to a direct, lexical equivalence, it is ranked

#### 4.4. EXPERIMENT 3: MATCHING USING BACKGROUND KNOWLEDGE<sup>69</sup>

highest in the result set.

2. The remaining matches, were ranked according to the number of DICE aspect taxonomies that supported the match
3. Matches based on the same number of DICE aspects, were ranked according to the number of equivalence matches on DICE properties (i.e. preferring equivalences over subsumptions).

We assess the performance of our method on the Gold Standard containing matches for a random set of 200 OLVG concepts, as we did for the baseline experiments. Having a ranked list for a single OLVG concept, we considered a suggested match correct if it was ranked highest in the list. When the ranking could not suggest a single best match, we considered the match correct if it was ranked in the top five and the ranking did not suggest other match as better<sup>11</sup>. This way the comparison revealed that the indirect matching approach produced 112 matches, out of which 107 were correct and 5 were incorrect. Hence, our method achieved **Recall** = 72% and **Precision** = 95%.

These figures are summarized in the table in Figure 4.11, including the results of the baseline experiments already reported in Section 4.3.2. The indirect matching not only increased the recall (found more correct matches), but also increased the precision as no additional incorrect matches are found as compared to the pure lexical matching.

This summary shows that on such semantically impoverished vocabularies as in our case study, the use of semantic background knowledge as part of the matching can substantially improve both the number of matches found and the quality of these matches, as compared to either a purely lexical technique, or existing tools that, in the absence of any structure in the source and target vocabulary, default to lexical matching only.

---

<sup>11</sup>In this context we have to choose what do we consider a correct match, because the method returns a ranked list which not always suggests one single best match. Given the use-case of this matching task, namely, to use the automatically-produced matches as a suggestion to the medical experts, if the suggested match is in the top five of the ranked list, that largely reduces the manual labor to a check of the first five suggested matches, as opposed to find the matching AMC concept in the entire vocabulary. Hence, when the correct match is suggested in the top five of the ranked list, it can be considered correct, as it basically solves the task it was created for. The reader is referred to [Van Rijsbergen, 1979] for more extensive discussion on relevancy of the retrieved results in the context of Information Retrieval

	<b>Recall</b>	<b>Precision</b>
<b>Indirect matching</b>	<b>72%</b>	<b>95%</b>
lexical matching	44%	93%
FOAM	28%	47%
Falcon-AO	19%	26%

Figure 4.11: Precision and recall obtained in all the experiments: The indirect matching outperformed the lexical matching and the state-of-the-art matching tools

## 4.5 Conclusions

We explored the use of a semantically rich background knowledge in matching semantically poor concept lists, and provided empirical evidence that background knowledge can boost the matching performance considerably as compared to the state-of-the-art matching tools. First, we showed that a simple lexical matching shows comparable performance to the lexical matchers in the state-of-the-art matchers on semantically poor data, and second, we showed that indirect matching about doubles the recall, at the same precision. The use of a background knowledge source is the only way to discover matches, when there is no terminological, instance or structural match between the matching ontologies.

Work by [Bouquet et al., 2003] has already shown the usefulness of simple background knowledge in the form of the WordNet hierarchy. In our case using a much richer source of background knowledge enabled us to reason across multiple hierarchies in the background knowledge, and made it possible to discover relations between concepts which were not directly related in a subsumption relation.

# Chapter 5

## Using multiple sources of background knowledge

The study in the previous chapter showed that using background knowledge can be more effective in matching semantically poor vocabularies than using direct matching. In this chapter we extend the investigation further, through two case studies we explore how the matching behaves when multiple background knowledge ontologies are used. We experimented using real-life ontologies as they exist today, and our experiments show that this approach works in practice and has great potential for solving the ontology matching problem. When adding more background ontologies in the matching, the recall increases considerably in a monotonic manner while the precision changes depending on the quality of the background knowledge - high quality results in high precision and low quality results in low matching precision.

### 5.1 Introduction

The benefit of using a background knowledge in ontology matching comes from its (rich) structure. The background knowledge is expected to enrich the joint structure of the matching ontologies, which then fosters the discovery of new matches that are otherwise missed in a direct matching.

On many places on the Internet ontologies are designed and published in commonly accepted formats, such as the RDF [Lassila et al., 1999] which originates from the Semantic Web [Berners-Lee et al., 2001]. Of particular interest are the ontologies published by specialists, as that gives confidence in the quality. Famous examples in the medical domain are the Foundational



Model of Anatomy [Rose and Jr., 2003] and the Unified Medical Language System [Bodenreider, 2004, D.A.B. Lindberg, 1993]. Furthermore, the open source communities are also actively contributing. In the music domain MusicBrainz<sup>1</sup> is one of the key players in providing music metadata, together with MusicMoz<sup>2</sup> which is part of a larger metadata collection and categorization initiative known as the Open Directory<sup>3</sup>.

However, abstracting our view from specific examples, we see that the available ontologies on the Internet show different quantitative and qualitative characteristics. The study of [Wang et al., 2006] which analyzes a corpus of 1300 ontologies gathered from the Internet, states that the ontologies published by the year 2006 vary in size with an unequal distribution. Only 19 ontologies in the corpus contained more than 2000 concepts, (which is about 1.5% of the entire corpus), and in terms of semantics, the study concludes, I quote: "the number of analyzed ontologies that make good use of OWL Lite features is less than 20% of the total number of OWL ontologies". Hence, concerning the size, few of the available ontologies are large while many are relatively small, and with respect to the expressiveness, small part makes use of expressive language constructs while the majority relies on a rather light-weight semantic expressiveness.

The increasing amount of available ontologies combined with the need for effective search initiated the development of ontology search engines. Among others, the currently active include: Swoogle<sup>4</sup>, Watson<sup>5</sup> and OntoSearch<sup>6</sup>. They offer search over the ontology content, and usually provide a complete information about the retrieved results: a downloadable version of the suggested ontologies, the entity inside the ontology that matched the search query, and other technical metadata like date of indexing, size and the source format<sup>7</sup>. The main functionality they offer is usually search through an interface similar to Google's<sup>8</sup>, where a keyword-based search query is submitted, and the feedback is list of ontologies which match the search query. As such, these search engines are good source for exploring and retrieving the existing ontologies.

From the ontology matching perspective, regardless of the choice, no back-

---

<sup>1</sup><http://www.musicbrainz.org/>

<sup>2</sup><http://musicmoz.org/>

<sup>3</sup><http://www.dmoz.org/>

<sup>4</sup><http://swoogle.umbc.edu/>

<sup>5</sup><http://watson.kmi.open.ac.uk/WatsonWUI/>

<sup>6</sup><http://www.ontosearch.com/En/srss.jsp>

<sup>7</sup>Other formats different than the standard RDFXML are also supported, Watson supports DAML for example

<sup>8</sup><http://www.google.com/>

ground knowledge ontology is likely to provide all the knowledge required for constructing a complete matching. Instead, it is reasonable to expect that the matches missed by one background ontology can be found using some other. Since multiple background ontologies can provide more matches than one ontology alone, with the aim to find a complete matching we can use *multiple ontologies as background knowledge*. Now, this seems like a very challenging task, especially given the discussion above that the existing ontologies are diverse in their characteristics. Exploiting multiple background ontologies poses the challenge how to integrate the reasoning over these different ontologies. In one of the simplest forms, each background knowledge can be used in a separate matching process, and then the obtained results can be combined to construct the required matching (like in combining aspects from DICE). In this chapter, we explore the use of multiple background ontologies based on this view.

In this chapter we set to investigate the feasibility of the matching when multiple background ontologies are used. To stress the paradigm, we present the results of a couple of experiments in which we set our objectives as follows: *(i)* the anchoring to the background knowledge is a simple lexical matching technique, i.e. we only use simple matching as needed to obtain relatively successful anchoring (see Section 4.3.1 for further explanation), *(ii)* the background knowledge candidates are relatively large sized ontologies<sup>9</sup>, and *(iii)* lexical anchoring from source and target to the background knowledge is possible, i.e. there is lexical overlap between the matching ontologies and the background knowledge.

We report two case studies. In the first, we carefully selected large and high-quality background knowledge ontologies. Intuitively, ontology can be considered high-quality if it is constructed by domain experts and it has been proven to be useful in practice<sup>10</sup>. In the second we used background ontologies of varying origin obtained through the Watson search engine.

Our investigation provides several crucial conclusions on the aspect of using multiple sources of background knowledge. The experiments show that the approach is successfully applicable in practice. Adding more background ontologies monotonically increases the recall, while the precision varies depending on the quality of the background knowledge.

The rest of the chapter is organized as follows: Section 5.2 describes

---

<sup>9</sup>Ontologies of size around 30 concepts are common as demonstration examples, however, they are trivial to analyze and do not provide well-grounded empirical insight. Hence, we focused our attention on ontologies of larger size with at least couple of hundreds of concepts as more interesting candidates.

<sup>10</sup>We note that no standard notion for ontology quality is available so far

the first case study which concerns matching medical ontologies while using large high-quality background ontologies, Section 5.3 describes the second case study which concerns matching agricultural ontologies using background knowledge of varying origin, and Section 5.4 concludes the chapter.

## 5.2 Case study 1

This case study extends on the one from the previous chapter, using the same matching data, but with extended background knowledge. Namely, we used two additional sources of background knowledge in the matching process. Again, the challenge here was to match the two medical vocabularies OLVG and AMC, but this time besides DICE we also used MeSH [Lipscomb, 2000, Lowe and Barnett, 1994] and ICD-10<sup>11</sup> ontologies as background knowledge. MeSH, is a large controlled vocabulary used for indexing publications in medicine, and ICD-10 is a classification of diseases and health related problems. We selected these two ontologies because of two reasons: first, they cover the same domain as the ontologies being matched, and the second and more important reason is that they are created and maintained by experts from the NLM<sup>12</sup> and WHO<sup>13</sup> organizations, which warrants a high quality. MeSH is actively used and improved for more than 50 years<sup>14</sup>, while the ICD family of disease classifications are around for more than a 100 years<sup>15</sup>.

For evaluating the matching performance in the experiments we created a Gold Standard reference matches for a randomly selected sample set of 200 from the 1399 OLVG concepts. The detailed description of the matching vocabularies, the DICE background knowledge and the process of creating the Gold Standard can be found in the previous chapter in Section 4.2.

**Matching ontologies** As source and target vocabularies, we used OLVG and AMC which are unstructured lists of concepts. OLVG contains 1399 and AMC 1460 concepts. These concepts describe reasons for admission, why a patient is brought in an Intensive Care Unit. More details in Section 4.2.

**Background knowledge ontologies** In the indirect matching experiments we used three ontologies as background knowledge. They were:

---

<sup>11</sup><http://www.who.int/classifications/icd/icdonlineversions/en/index.html>

<sup>12</sup><http://www.nlm.nih.gov/>

<sup>13</sup><http://www.who.int/>

<sup>14</sup>[http://www.nlm.nih.gov/mesh/intro\\_preface2007.html#pref\\_hist](http://www.nlm.nih.gov/mesh/intro_preface2007.html#pref_hist)

<sup>15</sup><http://www.who.int/classifications/icd/en/>

- *Background ontology 1: DICE* is an ontology that describes the domain of Intensive Care. Its core structure is organized in five hierarchies, each describing some aspect of the medical problems that cause patient to be brought in an Intensive Care Unit. DICE consists of 2300 concepts, of which 1920 constitute the five hierarchies that we used as a background knowledge. More details in Section 4.2.
- *Background ontology 2: MeSH (Medical Subject Headings)* is a large controlled vocabulary used for indexing biomedical and health-related journal articles and books. It is created and maintained by the United States National Library of Medicine (NLM), and is also used by the MEDLINE<sup>16</sup> article database and by NLM's catalog of book holdings. MeSH can be browsed and downloaded free of charge on the Internet<sup>17</sup>. At the highest level, MeSH distinguishes between 16 top level categories, which include *Anatomy, Diseases, Organisms, Technology and Food and Beverages, Geographic Locations, etc.*

For the purpose of our experiments we extracted the Dutch translation of MeSH from UMLS 2005AB [D.A.B. Lindberg, 1993] release. This version of MeSH contains a total of 22,568 subject headings, also known as descriptors. Most of these are accompanied by a list of synonyms. The descriptors are organized in a hierarchy, where a given descriptor may appear at several places in the hierarchy.

- *Background ontology 3: ICD-10 (International Classification of Diseases - tenth edition)* The International Statistical Classification of Diseases and Related Health Problems (commonly known by the abbreviation ICD) is a detailed description of known diseases and injuries. It is published by the World Health Organization<sup>18</sup> (WHO) and is used world-wide for morbidity and mortality statistics, reimbursement systems and automated decision support in medicine. The ICD is a core classification of the WHO Family of International Classifications (WHO-FIC). It is revised periodically and is currently in its tenth edition, known as the ICD-10. Annual minor updates and 3 yearly major updates are published by WHO.

In our experiments we used the Dutch translation of ICD-10, extracted from UMLS 2005AB [D.A.B. Lindberg, 1993] release. It contains 11,523 descriptions of diseases and injuries. They are structured in a single hierarchy. The concepts can be named with one or multiple terms,

---

<sup>16</sup><http://www.nlm.nih.gov/medlineplus/>

<sup>17</sup><http://www.nlm.nih.gov/mesh/>

<sup>18</sup><http://www.who.int/en/>

Background knowledge ontology	Size in number of concepts
DICE	1920
MeSH	22,568
ICD-10	11,523

Table 5.1: The size of each background knowledge ontology, expressed in number of concepts.

one chosen as a 'preferred' term - label, and the others are alternative synonyms.

The sizes of all the background knowledge ontologies are shown in the table on Figure 5.1. MeSH is largest of the three. It describes the entire medical domain, while DICE and ICD-10 are more focused, i.e. they describe medical problems.

**Evaluation: Gold Standard solution** For evaluating the performance of the different matching experiments we created a Gold Standard solution for this problem. In collaboration with a medical expert we created a reference matching on a sample of randomly drawn set of 200 OLVG concepts. Of these 200 concepts, we managed to find an AMC matching concept for 148, and left the other 52 unmatched. More details in Section 4.2.

### 5.2.1 Experiments

We conducted four experiments in which we matched the OLVG to the AMC vocabulary. The first experiment functioned as a baseline, and we used it as a reference to measure the matching improvement obtained by the use of different background knowledge ontologies. It used a simple lexical technique to match the vocabularies directly. In the three successive experiments (Exp.2 - Exp.4) we used background knowledge, one of the three ontologies described in the previous section (DICE, MeSH and ICD-10), per experiment. In the anchoring we used the same lexical technique used in the first (baseline) matching experiment.

**Lexical matching** OLVG and AMC were matched using a simple lexical technique based on a word matching. The method was presented in detail

in Section 4.3.1. Before the matching, in a preprocessing step the concept's labels were normalized. The matching was based on word comparison while also accounting for the Germanic constructions of compounding words together in a new meaningful forms, which are very common in the Dutch language.

**Matching using background knowledge** In the indirect matching experiments, we applied the algorithm described in Section 4.4. It finds indirect matches between the source and target vocabularies in two steps. In the first step, anchoring, the source and target concepts are anchored to the background knowledge, and based on how the anchored concepts in the background knowledge are related, the indirect matches are induced in a second step - deriving relations. Single source or target concept can have multiple anchors in the background knowledge enabling the discovery of multiple indirect relations through the anchored concepts, which then reinforces the (indirect) match.

### **Experiment 1: Lexical matching**

When applied the lexical technique produced 582 matches in directly matching OLVG to AMC. Of these, 274 were found to be lexically equivalent concepts, and the remaining 308 were partial lexical matches. The details of this experiment were presented in Section 4.3.1.

### **Experiment 2: DICE as background knowledge**

As a result of applying DICE as a background knowledge, a matching AMC concept was derived for 548 OLVG concepts. Of these, 413 matches were based on inference in a single DICE aspect, while 135 matches were supported by inference in two aspects. The details of this experiment were presented in Section 4.4.

### **Experiment 3: MeSH as background knowledge**

Manual inspection of MeSH revealed that it has 16 top level categories. We selected the most relevant as separate aspects and collected the others together. We used: *Analytical, Diagnostic and Therapeutic Techniques and Equipment* as the first aspect, *Anatomy, Diseases and Biological Sciences* as the second, third and fourth respectively, and the remaining categories together as the fifth aspect.

	OLVG $\mapsto$ MeSH	AMC $\mapsto$ MeSH
<i>Therap. Techn. and Equip.</i>	242	315
<i>Anatomy</i>	517	945
<i>Diseases</i>	914	1412
<i>Biological Sciences</i>	161	228
<i>Other</i>	352	454
	1916	3089

Table 5.2: Distribution of anchors over the different MeSH taxonomies

**Anchoring** When anchoring the OLVG vocabulary to MeSH using the lexical technique, we were able to anchor 1061 of the 1399 OLVG concepts, using 1916 anchors in total. For the AMC list, we found 3089 anchors, together anchoring 1277 of the 1460 concepts.

**Deriving relations** The relatively high number of anchors for MeSH also resulted in a large number of matches. We found 298 matches that were based on two or more aspects (i.e. inference in two or more MeSH taxonomies produced support for the same match), and another 694 based on only one aspect. When we compare this to the results achieved with the DICE ontology (see Table 5.3 for an overview), we see that the matches found using MeSH have more support than those found using DICE as background knowledge. This makes a very important conclusion that, namely, the expert manually-created anchors used in DICE are not an explanation for its matching performance.

#### Experiment 4: ICD-10 as background knowledge

ICD-10 contains a single hierarchy of diseases and injuries, and does not distinguish different hierarchies of anatomy or diseases. Therefore, in our experiment we considered it to be one single aspect.

**Anchoring** When anchoring the OLVG to ICD-10, 134 of the OLVG concepts were anchored to concepts in ICD-10, via 144 anchors. From the AMC list, 207 concepts were anchored, via 222 anchors.

An analysis to explain this low number of anchors revealed that by improving the lexical anchoring algorithm an additional 12% could be produced on the remaining non-anchored OLVG and AMC concepts. Some concepts,

	DICE	MeSH	ICD-10
4 aspects	0	8	0
3 aspects	0	89	0
2 aspects	135	201	0
1 aspect	413	694	80
Total	548	992	80

Table 5.3: Amount of derived matches supported by different number of aspects, for each of the ontologies used as background knowledge.

were impossible to anchor because there was no counter-part in ICD-10; **Heroin** for example, does not occur in any concept in the Dutch translation of ICD-10. The major reason for the anchoring failure was the different granularity of ICD-10. The concepts in ICD-10 are described in a precise way, usually using more than five words in the descriptions, and high-level concepts such as **Alcohol** or **Drugs** are not present. As a consequence, it is hard to anchor the **OLVG** and **AMC** concepts to the ICD-10 ontology. Important to note here is that this is not a problem of using too simplistic anchoring technique, it is rather a semantic mismatch. In most of the cases the matching **OLVG** and **AMC** concepts can not be appropriately described using ICD-10 concepts.

**Derived Matches** In the matching phase, only 80 **OLVG** concepts were matched to the **AMC** list. One reason for such low matching performance was the low amount of anchors established in the first step. The other reason is that the semantic structure that ICD-10 provides is much in line with the lexical correspondence between its terms. As a consequence, when used as background knowledge, the ICD-10 ontology did not improve the lexical matching.

## 5.2.2 Evaluation

The evaluation of the experimental results against the Gold Standard is made more complicated by the fact that many-to-many anchors can also produce many-to-many matches between source and target vocabularies. When a single **OLVG** concept matches with multiple concepts in the **AMC** vocabulary, we ranked the matches as follows:



1. If the match is detected directly using the lexical technique, it is considered as the most reliable answer, and is accordingly ranked highest in the result set.<sup>19</sup>
2. The remaining matches were ranked according to the number of different aspect taxonomies that supported the match
3. Matches based on the same number of aspects were ranked according to the number of equivalence matches within each aspect (i.e. preferring equivalences over subsumption).

We assess the performance of the different experiments on the Gold Standard containing matches for a random set of 200 OLVG concepts, as we did for the experiments in the last chapter. Having a ranked list for a single OLVG concept, we considered a suggested match correct if it was ranked highest in the list. When the ranking could not suggest a single best match, we considered the match correct if it was ranked in the top five and the ranking did not suggest other match as better. This assessment produced the following results:

- The lexical matching technique produced 70 matches, 65 correct and 5 incorrect.
- Using DICE as background knowledge produced 112 matches, 107 correct and 5 incorrect.
- Using MeSH as background knowledge produced 130 matches, 116 correct and 14 incorrect.
- Using ICD-10 as background knowledge produced 70 matches, 65 correct and 5 incorrect. The use of ICD-10 did not change the result from the lexical matching.

Next, we evaluate the performance of these matchings by calculating the precision and recall for each of the matching experiments (as described in Section 3.3.1). The results are presented in table in Figure 5.1.

---

<sup>19</sup>The direct lexical matches between the OLVG and AMC vocabulary, i.e. the results of our baseline experiment, are also included in the results of the experiments with background knowledge.

	<b>Recall</b>	<b>Precision</b>
Exp.1 Lexical matching	44%	93%
Exp.2 DICE	72%	95%
Exp.3 MeSH	78%	89%
Exp.4 ICD-10	44%	93%

Figure 5.1: Precision and recall obtained in all the experiments.

### 5.2.3 Analysis

As to be expected, there are differences in performance between the various knowledge sources: **MeSH** when used as background knowledge outperformed the others in terms of recall. Even though with a lower precision than **DICE**, increasing the recall, i.e. finding more correct matches, in our use-case can be considered a better improvement, because checking if a match is correct or incorrect takes less time than finding the corresponding match in the target.

The table in Figure 5.1 shows that as compared to lexical matching, the recall is improved when **DICE** and **MeSH** are used as background knowledge, and it remains the same in the case of **ICD-10**. The precision is slightly improved in the case of **DICE** and drops for **MeSH**. Importantly, the recall is increased, and as discussed in Section 3.4 it is currently a major challenge in the ontology matching problem.

**Cumulative use of multiple sources:** The different background knowledge sources do discover different matches. For example, using **MeSH** a match

OLVG: Dunne darm resectie → AMC: Ileum resectie

was found which was not found when using **DICE**. On the other hand, using **DICE**, we found the match

OLVG: Heroine intoxicatie → AMC: Drugs Overdosis

which was not found using **MeSH** (because the concept **Heroine** is not present in the Dutch translation of **MeSH**).

This suggests that it is indeed useful to use the background sources cumulatively. Using multiple sources, we will find more matches. Further analysis showed that the combination of **DICE** and **MeSH** resulted in 122 correct matches on the sample set from the Gold Standard which is a recall of 82%, or an improvement of 38% over the lexical matching baseline (as compared to improvements of 28% for **DICE** and 34% by **MeSH** separately, as

shown in Figure 5.1), and it resulted in 90% precision. Adding ICD-10 to the collection does not improve the results any further (as can be predicted from Figure 5.1), but importantly, adding ICD-10 does not hurt the matching, i.e. it does not make the results any worse because it did not contribute any new matches. In other words, if an expert-created background knowledge, as in this case, provides an answer then it is probably the correct one<sup>20</sup>. Thus, our case-study reveals an attractive monotonic relation between the amount of extra background knowledge that is used and the quality of the matches that are being discovered.

Notice that the different background sources are each used independently of each other. This has as advantages that: the resulting matches are simply the union of those found by using the individual knowledge sources, and the computational costs rise only linearly with the number of background sources (no further expensive interactions between knowledge in the different sources is required).

#### 5.2.4 Conclusions of case study 1

We conclude that multiple high-quality background knowledge ontologies can be successfully used cumulatively in the ontology matching process. Each of the background ontologies provides a different contribution, and their cumulative contribution results in a monotonic increase of the matching performance.

As intuitively expected, the size of the background knowledge is one key factor for a successful matching. The experiments of using MeSH showed that large size of the background knowledge increases the chances for successful matching. However, it does not necessarily guarantee the improvement, as shown in the experiment with ICD-10.

Furthermore, the experiment with DICE indicated that the content overlap with the source and the target ontologies plays an important role in the matching performance. DICE provides coverage of the intensive care domain, and although it is smaller than MeSH and ICD-10, it considerably improved the results as compared to the baseline matching. Hence, we observe another success factor, which is the lexical overlap with the source and the target ontologies.

---

<sup>20</sup>This behavior is also dependent on the high precision in the anchoring, namely, in the experiments we used a very rigid anchoring technique, however, if there are false anchors established then it is likely that the background knowledge will start producing wrong indirect matches as well.

## 5.3 Case study 2

In this case study we matched two ontologies from the agricultural domain using six other ontologies as background knowledge. Motivated by the variety of ontologies that exist online, we decided to use background knowledge ontologies with varying origin. We investigated three different types of ontologies: ontologies that model domain different from the domain of the ones being matched, general knowledge ontologies, and ontologies of an unknown origin. Similarly as in the previous case study, we set simple direct matching as a baseline to evaluate the matching performance, and we analyzed the matching performance by observing the precision and recall.

**Matching ontologies** The source ontology was NALT<sup>21</sup> and the target *Agrovoc*<sup>22</sup>. They both describe the agriculture domain. *Agrovoc*, as stated on the description provided on its homepage<sup>23</sup>, I quote "is a multilingual, structured and controlled vocabulary designed to cover the terminology of all subject fields in agriculture, forestry, fisheries, food and related domains (e.g. environment)." NALT, as described on its homepage<sup>24</sup>, I quote: "The NALT is primarily used for indexing and for improving retrieval of agricultural information. Currently, the NALT is the indexing vocabulary for NAL's bibliographic database of citations to agricultural resources, AGRICOLA<sup>25</sup>. The Food Safety Research Information Office<sup>26</sup> (FSRIO) and Agricultural Network Information Center<sup>27</sup> (AgNIC) also use the NALT as the indexing vocabulary for their information systems. In addition, the NALT is used as an aid for locating information on the ARS<sup>28</sup> and AgNIC web sites." In the experiments we used the versions of the OAEI 2006<sup>29</sup>, which are publicly available. They contain 41,577 concepts NALT, and 28,174 concepts *Agrovoc*. Many of the concepts besides the labels are additionally described with synonyms.

**Background knowledge** We selected the background knowledge ontologies to faithfully represent the types of background knowledge we set to

---

<sup>21</sup><http://agclass.nal.usda.gov/agt>

<sup>22</sup><http://www.fao.org/agrovoc>

<sup>23</sup>[http://www.fao.org/aims/ag\\_intro.htm](http://www.fao.org/aims/ag_intro.htm)

<sup>24</sup><http://agclass.nal.usda.gov/about.shtml>

<sup>25</sup><http://agricola.nal.usda.gov/>

<sup>26</sup><http://www.nal.usda.gov/foodsafety/>

<sup>27</sup><http://www.agnic.org/>

<sup>28</sup><http://www.ars.usda.gov/>

<sup>29</sup>Published on <http://www.few.vu.nl/~wrvhage/oaei2006/>

investigate. As said, we used the Watson<sup>30</sup> ontology search engine to find them. We queried Watson for concept labels from the matching ontologies which are common English terms like *meat*, *animal*, *food*, etc. and selected six ontologies which frequently occurred in the retrieved results and also seemed like reasonable choice for the goal we set to analyze, that is exploring the different background knowledge types. Note that the choice of the search engine is not any special, in other studies different search engines have been successfully used for the same purpose, [Sabou et al., 2006] used Swoogle to dynamically select background ontologies for an ontology matching task.

The selected six ontologies were the following: **Economy** which models a different but related domain as the matching ontologies; **Mid-level**, **Sumo** and **Tap** which are general knowledge ontologies; and **A.com** and **Surrey** which are ontologies of an unknown origin.

- *Background knowledge 1*: **Economy** ontology is described at [www.daml.org](http://www.daml.org), I quote: "is based on CIA World Fact Book (2002). Some industry concepts are based on the North American Classification System ('NAICS') - online at <http://www.census.gov/rpcd/www/naics.html>." As its name indicates, it intends to formally describe the domain of economy. It was engineered by Teknowledge Corporation<sup>31</sup> and submitted to the collection of ontologies gathered at [www.daml.org](http://www.daml.org). The size is 323 concepts.
- *Background knowledge 2*: **Mid-level** is constructed to play the role of bridge between the **Sumo** abstract level ontology, and the different varieties of Sumo domain-specific ontologies<sup>32</sup>. It is not domain-specific, and contains 1773 concepts.
- *Background knowledge 3*: **Sumo** (Suggested Upper Merged Ontology) is being created as part of the IEEE Standard Upper Ontology Working Group. It contains 576 concepts.
- *Background knowledge 4*: **Tap** as described in [Guha and McCool, 2003] is a shallow but broad knowledge base containing basic lexical and taxonomic information about a wide range of popular objects. It is claimed to be independent of a domain, however, a manual inspection indicated that it mainly covers the chemical, machine and electronic industry domains. It contains 5488 concepts.

---

<sup>30</sup><http://watson.kmi.open.ac.uk/WatsonWUI/>

<sup>31</sup><http://www.teknowledge.com/>

<sup>32</sup><http://ontology.teknowledge.com/>

- *Background knowledge 5*: A.com is an ontology with an unknown origin. By browsing it we got the impression that it has been produced as a result of merging several ontologies. In addition, noticeable are surprising relations such as:

Volume  $\preceq$  Pollution

which can be seen as an indication that some form of directory structure was the origin of the data. It seems to cover various domains, and its size is 5624 concepts.

- *Background knowledge 6*: Surrey ontology, according to the Watson search engine, originates from the web site [www.surrey.co.uk](http://www.surrey.co.uk). In our analysis we did not manage to trace back its source, the download link does not work and on the web site the ontology is not available. Similarly as in the previous case, parts of its content gave the impression that it was created by transforming a directory structure into an ontology in a straight-forward way. Having no available documentation about how it was created, we treated it as an unknown origin ontology. Its size is 672 concepts.

Background knowledge ontology	Type of ontology	Size in number of concepts
BK <sub>1</sub> : Economy	Different domain	323
BK <sub>2</sub> : Mid-level	General knowledge	1773
BK <sub>3</sub> : Sumo	General knowledge	576
BK <sub>4</sub> : Tap	General knowledge	5488
BK <sub>5</sub> : A.Com	Unknown origin	5624
BK <sub>6</sub> : Surrey	Unknown origin	672

Figure 5.2: Properties of the background knowledge ontologies

The six background knowledge ontologies with their properties are summarized on the table in Figure 5.2. Their sizes are smaller than the sizes of the background ontologies used in the case study 1 (see the table in Figure 5.1), but with respect to the common ontology sizes found online [Wang et al., 2006], they are large sized ontologies.

**Evaluation** We manually evaluated the results of the matching experiments. As a reference use-case we set the task of document reclassification, which is realistic in this context because the matching ontologies are used for classifying books and articles.

```

    The set of direct matches is empty in the beginning
1   dmatches :=  $\emptyset$ 

    Lexical phase: find equivalent lexical matches
2   for every concept pair  $X \in \mathcal{C}^{\text{SRC}}, Y \in \mathcal{C}^{\text{TAR}}$  do
3     if FULLLEXMATCH( $X, Y$ ) then
4       dmatches  $\leftarrow (X \stackrel{\equiv}{\rightarrow} Y)$ 
5   end for

    Structural phase: use the structure to find more matches
6   for every two relations  $(X_1 \preceq X_2) \in \mathcal{R}^{\text{SRC}}, (Y_1 \preceq Y_2) \in \mathcal{R}^{\text{TAR}}$ 
7     if  $(X_2 \stackrel{\equiv}{\rightarrow} Y_1) \in \text{dmatches}$  then
8       dmatches  $\leftarrow (X_1 \stackrel{\preceq}{\rightarrow} Y_2)$ 
9   for every two relations  $(X_1 \succeq X_2) \in \mathcal{R}^{\text{SRC}}, (Y_1 \succeq Y_2) \in \mathcal{R}^{\text{TAR}}$ 
10  if  $(X_2 \stackrel{\equiv}{\rightarrow} Y_1) \in \text{dmatches}$  then
11  dmatches  $\leftarrow (X_1 \stackrel{\succeq}{\rightarrow} Y_2)$ 

```

Figure 5.3: Algorithm for matching ontologies directly.

### 5.3.1 Experiments

We performed seven experiments in which we matched NALT to Agrovoc. In the first experiment, which served as baseline, we matched the ontologies directly, and in the other six we matched them indirectly using the six previously described background ontologies, one at a time.

**Direct matching (Experiment 1)** In the direct matching we combined lexical and structural matching. In the lexical phase the labels were normalized by discarding stop words and interpunction, and then matched to one another accounting for different word order and plural/singular form of the words. As a result, the lexical phase produced list of pairs of equivalent concepts. In the structural phase the hierarchical structure of the ontologies was used to induce further matches. The direct matching algorithm is shown in Figure 5.3.

When applied, the direct matching produced 6,437 matches between NALT and Agrovoc. This number is comparable to the numbers obtained in the OAEI 2006 [Euzenat et al., 2006] on the same test data, where most of the participating matching systems produced between 5000 and 10,000 matches.

```

    The set of indirect matches is empty in the beginning
1  imatches :=  $\emptyset$ 

    Anchoring phase: anchor SRC and TAR to BK using direct matching
2  anchS $\mapsto$ B := MATCHDIRECTLY(SRC, BK)
3  anchT $\mapsto$ B := MATCHDIRECTLY(TAR, BK)

    Deriving relations phase: find indirect matches using the anchors and BK
4  for every two anchors  $(X \mapsto Z_1) \in \text{anch}^{S \mapsto B}, (Y \mapsto Z_2) \in \text{anch}^{T \mapsto B}$ 
5    if  $(Z_1 \preceq Z_2)$  then
6      imatches  $\leftarrow (X \mapsto Y)$ 
7  for every two anchors  $(X \mapsto Z_1) \in \text{anch}^{S \mapsto B}, (Y \mapsto Z_2) \in \text{anch}^{T \mapsto B}$ 
8    if  $(Z_1 \succeq Z_2)$  then
9      imatches  $\leftarrow (X \mapsto Y)$ 

```

Figure 5.4: Algorithm for matching SRC to TAR indirectly through BK as a background knowledge.

**Indirect matching (Experiments 2 - 7)** In the indirect matching we lexically anchored the matching ontologies to the background knowledge, and then used the hierarchies of the background knowledge to induce the indirect matches. In other words, the indirect matching algorithm can be explained as follows: for two matching concepts we first find their equivalent concepts in the background knowledge (if possible), then check if these background concepts are hierarchically related, and if they are we report an indirect match between the matching concepts. The indirect matching algorithm is shown on Figure 5.4.

Background knowledge	BK <sub>i</sub> size	Source anchors	Target anchors
BK <sub>1</sub> : Economy	323	121	106
BK <sub>2</sub> : MidLevel	1773	330	271
BK <sub>3</sub> : Sumo	576	79	72
BK <sub>4</sub> : Tap	5488	367	227
BK <sub>5</sub> : ACom	5624	66	69
BK <sub>6</sub> : Surrey	672	102	95

Figure 5.5: Overview of the anchoring results.

The table on Figure 5.5 summarizes the results of the anchoring phase showing the number of source and target anchors (NALT and Agrovoc, respectively) established to each background knowledge ontology. The Economy ontology has the highest number of anchors as compared to its size, roughly



to about one third of its concepts there are anchors established from the matching ontologies. Contrary, ACom has much fewer anchors relatively to its size, roughly one out of each 90 concepts has anchor established to it. We can also observe from the table that this ratio is variable for the background ontologies of the same origin.

Generally, the number of anchors is much smaller than the sizes of the matching ontologies NALT and Agrovoc, which count in tens of thousands. However, given the sizes of the background ontologies and the fact that they are not agriculture-specific, this anchoring result is not surprising.

Background ontology	BK <sub><i>i</i></sub> size	Indirect matches	Additional matches on top of direct matches
BK <sub>1</sub> : Economy	323	259	85
BK <sub>2</sub> : MidLevel	1773	200	81
BK <sub>3</sub> : Sumo	576	115	57
BK <sub>4</sub> : Tap	5488	1003	625
BK <sub>5</sub> : ACom	5624	87	71
BK <sub>6</sub> : Surrey	672	623	543
<b>Cumulatively all BK<sub><i>i</i></sub></b>		2183	1428

Figure 5.6: Overview of the indirect matching results, the number of matches established using each background ontology

The table in Figure 5.6 gives an overview on the indirect matching results. The third and fourth column show the number of indirect matches, and the number of additional indirect matches which were not found in the baseline direct matching. Each row in the table corresponds to one background knowledge ontology, except for the last one which shows the *cumulative* number of matches (union). Note that these cumulative numbers are not simple sums of the numbers above them, for example for the indirect matches the sum is 2287 and the cumulative number of matches is 2183. They are different because some of the matches are found by more than one background knowledge ontology. Similarly, the sum of the additional indirect matches is 1462 whereas the cumulative number is 1428. We see that the sum and the cumulative number are close to one another, which reveals very important and attractive behavior of using multiple background knowledge ontologies, namely, different ontologies produce nearly disjoint sets of indirect matches. This means that the more ontologies we use - the more matches we will find. If we look at the cumulative matches, the additional indirect matches represent 66% of all the indirect matches, which in turn means that an arbitrary

indirect match has higher chances to be an addition to the baseline matches. However, these numbers say nothing about the quality of the matches, as a next step we will evaluate their correctness.

### 5.3.2 Evaluation

In order to get better insight in the matching process we decided to undertake the effort of manually assessing the matches. As a natural reference we choose the task of document reclassification: the obtained matches are expected to faithfully reclassify the documents from the source to the target ontology, ideally, in the same way as a human would do.

For the precision we did the evaluation as follows: each match was checked for validity, if the correctness was not obvious then Google was used as reference by querying for *define: label* to find the definition of the term *label*. The evaluation of the precision proceeded in two phases: first evaluate the direct and then the indirect matches. For the direct matching which produced more than 6000 matches, we choose the random sampling method. After drawing a random sample of 10% (640 matches), we manually assessed these matches as described above. For the indirect matches, which were in total little bit more than 2000, we took the effort to manually assess all of them.

The recall was hard to estimate because it requires all the correct matches between the matching ontologies available, which we don't have. Therefore we set to observe the change in recall between different experiments instead of estimating the achieved recall.

The evaluation revealed that the direct matching achieved 100% precision, i.e. all the matches in the evaluation sample were correct. The precision of the indirect matching and the change in recall are shown in the table on Figure 5.7.

### 5.3.3 Analysis

First general observation on the matches (all the matches from all the seven experiments) is that they were established between a small subset of the matching concepts: 2241 in NALT, and 1757 concepts in Agrovoc participated in the matches, as compared to the size of NALT which is 41,577 concepts and Agrovoc 28,174 concepts. The number of concepts which participated in the matching results were in the order of about 5% of the size of the matching ontologies. But, this effect is not peculiarity of our experimental data, in other studies [Zhang and Bodenreider, 2003, Mork et al., 2004]

Matching experiment	<b>Precision</b>	<b>Precision</b>	<b><math>\Delta</math>Recall</b>
	indir. matches	addit. matches	
Exp.2: BK <sub>1</sub> : Economy	84.17%	51.76%	0.68%
Exp.3: BK <sub>2</sub> : Mid-level	97.00%	92.59%	1.17%
Exp.4: BK <sub>3</sub> : Sumo	76.52%	52.63%	0.47%
Exp.5: BK <sub>4</sub> : Tap	57.23%	31.36%	3.04%
Exp.6: BK <sub>5</sub> : A.Com	36.78%	22.54%	0.25%
Exp.7: BK <sub>6</sub> : Surrey	35.63%	26.15%	2.21%
<b>Cumulativly BK<sub>1</sub>-BK<sub>6</sub></b>	<b>57.63%</b>	<b>35.22%</b>	<b>7.81%</b>

Figure 5.7: Performance of the indirect matching experiments

similar effect was noticed when matching the **FMA** and **GALEN** ontologies which model the human anatomy. These ontologies have 59,000 and 24,000 concepts respectively, and the number of matched concepts reported in the studies is in the order of 10% of the ontology sizes. It seems that this effect occurs when matching large ontologies even though they model the same domain. Most likely explanation for this is that for the general concepts there is much better naming agreement, while for the more specific ones, which represent the majority, there is almost no agreement. In such a situation the labeling problem is solved by using many words to name a single concept. As an example, in **NALT** there is a concept named *Salmonella choleraesuis* subsp. *choleraesuis* serovar *Paratyphi A*.

**Discussion on the precision and recall** The table on Figure 5.7 shows the precision of the indirect matches, and the increase of recall with respect to the baseline direct matching. Each row corresponds to one background ontology, except for the last which shows the results for the cumulative use of all the background ontologies together.

All the indirect matches which were also found in the baseline were correct, incorrect matches only appeared when they were not found in the baseline matching. Hence, the precision of the additional indirect matches is lower than the precision of the indirect matches.

The **Tap** ontology resulted in 57.23% precision, however, a special situation had reduced the precision of this ontology. Many of the matches were wrongly established to the target concept called **Node**. The root concept in **TAP** is called **Node**, and the target concept anchored to it was found related to any source concept anchored in **Tap**. When these wrong matches are not

taken into account, the precision of **Tap** is calculated to 92.13%. This example gave a very important insight, the indirect matching can be very sensitive to mistakes which are high in the background knowledge hierarchy. The fact that the root concept of **Tap** was named **Node** caused drastic change in the results when we used it as background knowledge.

The first four background ontologies which are expert-created exhibit high precision in the indirect matches (more than 75%), and relatively high precision in the additional indirect matches (more than 50%). This reinforces the findings from the case study 1 that the high-quality ontologies show high precision. On the other hand, the unknown-origin ontologies show lower precision which is not a surprising thing given the low quality of their content.

Observing the recall we see that **Tap** provides the highest increase of recall, shown in the third column, but the **Surrey** ontology is the second next to the **Tap** ontology in the recall increase. While the ontologies of an unknown origin might show low precision, that does not prevent the recall being increased considerably. We also see that **Surrey** is much smaller than **Midlevel**, **Tap** and **ACom**, which is an empirical proof that small size does not immediately imply low recall.

**Causes of wrong matches** For the first four background knowledge ontologies there were two main causes for wrong matches: contextual problems and relatively small mistakes. Examples of matches caused by contextual problems are the following:

NALT: Meat	$\xrightarrow{\lambda}$	Agrovoc: Product
NALT: Vehicle	$\xrightarrow{\lambda}$	Agrovoc: Product
NALT: Organism	$\xrightarrow{\lambda}$	Agrovoc: Agent

Meat can be seen as a kind of product in the domain of economy, however, for our matching task this was not a desirable match. These matches can be seen as relations establishing roles, meat and vehicles can have the role of a product, and organism can have the role of an agent. Such modeling is apparently good for the contexts of these background ontologies. For discussions related to the context issues in knowledge representation the reader is referred to the Cyc Knowledge Base [Lenat, 1995] and the study of [Davis et al., 1993]<sup>33</sup>. In

---

<sup>33</sup>The study argues that the knowledge representation issues and the functionality of the system are intrinsically tied to one another, I quote: "Representation and reasoning are inextricably intertwined: we cannot talk about one without also, unavoidably, discussing the other. We argue as well that the attempt to deal with representation as knowledge content alone leads to an incomplete conception of the task of building an intelligent reasoner."

addition to the context problems, few of the wrong matches were caused by relatively small mistakes, such examples are the matches:

$$\begin{array}{l} \text{NALT: Marine invertebrae} \xrightarrow{\lambda} \text{Agrovoc: Fish} \\ \text{NALT: Herbivore} \xrightarrow{\lambda} \text{Agrovoc: Mammals} \end{array}$$

Jellyfish are kind of Marine invertebrae but they are not fish, and some kinds of birds are herbivore but not mammals. These relations come close to generally accepted claims like "birds fly" while exceptions exist: "penguins are birds, and yet they do not fly". We stress here that there were no different causes for wrong matches between **Economy** and the other three general-knowledge ontologies. The high-quality ontologies, whether they model different domain or are general-knowledge, the same reasons caused them to produce wrong matches when they were applied as background knowledge.

For the last two ontologies, which have unknown origin, mistakes were the cause for the wrong matches. For example:

$$\begin{array}{l} \text{NALT: Gas} \xrightarrow{\lambda} \text{Agrovoc: Turbines} \\ \text{NALT: Waste} \xrightarrow{\lambda} \text{Agrovoc: Water} \end{array}$$

The concepts in these wrong matches are semantically related, however, no strict relation can be established. These matches are clearly wrong. This suggests that **ACom** and **Surrey** were obtained by straight-forward transformation of a directory structure into an ontology.

### 5.3.4 Conclusions of case study 2

Observing the precision, the expert-created ontologies such as **Economy**, **Mid-level**, **Sumo** and **Tap** resulted in relatively high precision (more than 75%), and the main causes of wrong matches were contextual differences with the matching ontologies and small mistakes. The ontologies of unknown origin like **ACom** and **Surrey** resulted in lower precision (less than 40%) and the main cause of wrong matches were mistakes. This makes the expert-created ontologies more trustworthy and clearly preferable background knowledge candidates over the unknown-origin ontologies with respect to the precision.

All the background ontologies together provided relatively small increase in the recall of about 8% in addition to the direct matching. However, they resulted in nearly disjoint sets of matches, which means that new ontologies are likely to provide new additional matches and further increase the recall.

Furthermore, the expert-created ontologies, regardless whether they modeled different domain from the matching ontologies (**Economy**) or they were

general-knowledge (Mid-level, Sumo and Tap), they resulted in similar matching qualities. On our experimental data we could not discriminate by the precision or the recall increase, and all of them had the same causes of wrong matches.

Finally, the Tap ontology showed that the matching process can be very sensitive to mistakes high in the background knowledge hierarchy. Other mistakes also resulted in wrong matches, but the mistake in Tap with the root concept being labeled Node seriously affected the precision when applying this background ontology.

## 5.4 Conclusions

The question of how successful is using multiple background ontologies in the matching is made more complicated by the fact that different applications require different qualities of the matching system. Importantly, both case studies showed that the recall monotonically increases with adding more background ontologies. This is an important property because increasing the recall is seen as bigger challenge in the current matching systems. For the precision, the success depends on the qualitative characteristics of the background ontologies.

In the first case study the recall is almost doubled as compared to the baseline matching (from 44% to 82%). The ontologies provide large and to some extent overlapping contribution, i.e. the same matches are discovered by more background ontologies. In the second case study the recall increased 8%, which is smaller contribution, but unlike the first case study, in the second the contributions were disjoint, almost no matches were found by more than a single ontology. This encourages adding more background ontologies because the experiments indicate the recall will increase further. Interestingly, the second case study showed that not only the expert-created ontologies increase the recall, but the ontologies of unknown origin can also provide recall increase comparable to what expert ontologies do.

Both case studies showed that large ontology size can result in large contribution to the matching, but that is not necessarily the case. In the first case study DICE, and in the second the Surrey ontology contributed more than other ontologies even though they were smaller in size.

We conclude that the precision depends on the quality of the background ontologies and the context of the knowledge they model. The extensive expert-created ontologies in the first case study, that precisely described

the domain of the matching ontologies achieved precision around 95%. The expert-created ontologies in the second case study showed lower precision which was above 75%, where the precision was mainly reduced due to the context of the knowledge they model. Finally, the ontologies of unknown origin showed low precision below 40%, and this was mainly caused by mistakes in their content. Another important lesson about the precision was the experiment with the **Tap** ontology, which pointed out that precision can be very sensitive to mistakes high in the background knowledge hierarchy.

## Chapter 6

# Combining background knowledge relations

In the previous chapters we discussed the use of background knowledge in matching ontologies. In the reported experiments, the reasoner mainly made use of the subsumption hierarchies present in the background ontologies. In this chapter we will investigate the reasoning process further, in particular combining relations of different types from the background knowledge in a matching task. We will provide two case studies. In the first, we integrate two anatomy ontologies using a comprehensive and structure-rich domain ontology as background knowledge. The study suggests that largest benefit in the matching is obtained when the different relation-types in the background knowledge are combined, but obtaining this benefit requires prior analysis to determine how to combine the relations in the background knowledge. In the second case, we use a music ontology to mediate the process of matching user's preferences with classes in music provider's classification schemes. This enables the users to use a richer vocabulary in expressing their preferences, which is more natural than the classical search scheme offered by the music content providers. Both case studies were conducted using existing available ontologies. Combining different background relations appears to be useful, and perhaps even needed for a successful matching in these applications. This chapter is based on the work published in [Aleksovski et al., 2006c, Aleksovski et al., 2006b].



## 6.1 Introduction

The subsumption hierarchies provide a natural way of organizing the concepts in a domain<sup>1</sup>. They relate the concepts by stating that one is more or less general than another concept. The currently active OWL web ontology standard [Horrocks et al., 2003] inherits the subsumption expressivity from simpler formats such as the RDFS<sup>2</sup>. The existing ontologies, however, go beyond the subsumption hierarchies and use other relations between the concepts to express domain specific knowledge.

The *medical domain* is rich with ontologies, in particular expert-created [Rose and Jr., 2003, Bodenreider, 2004, Bodenreider et al., 1998, Cimino et al., 2003], that describe the human anatomy, diseases, medications, etc. Anatomy ontologies, such as the FMA<sup>3</sup>, organize the different parts of the human body in hierarchies, but also relate them with medical-specific relations, expressing that, for example, one part is attached to another, or one organ sends output to another. In anatomy, besides subsumption, *part-of* is widely used to build hierarchies which are perhaps even more useful systematic organization of the vast amount of different anatomical parts. *Part-of*, even though is generic and possible to encounter in many domains, it is still domain and context dependent, [van Hage et al., 2006].

The *music domain* ontologies organize the metadata that describes music content, and other music-related entities. It is an interesting commercial domain which suffers from ill-defined conceptualization, but contains a lot of formalized metadata collections used in various ways for finding music of interest. The music content is generally organized in classes such as genres and styles, but is also described through other music-specific attributes such as instruments, moods, tempos, rhythms, etc. This metadata is becoming crucially important for organizing and finding music, given the amount and the increasing pace of available music content.

The above indicates that ontologies rich with domain specific relationships are increasingly becoming available. Then, a natural question is, how can we benefit from this knowledge in ontology matching?

Domain specific relations have already been used in some of the existing matching tools. Falcon-AO [Jian et al., 2005], for example, interprets any relation between concepts as a kind of similarity, and performs some form

---

<sup>1</sup><http://www.w3.org/TR/owl-features/>

<sup>2</sup><http://www.w3.org/TR/rdf-schema/>

<sup>3</sup>*Foundational Model of Anatomy*, a description is provided in Section 6.2.1

of similarity flooding through the structure of the matching ontologies (intuitively, related concepts are more likely to be similar than the unrelated, no matter the kind of relatedness), see Section 2.1.2 for more details. In the current investigation we choose not to ignore the specific semantics of the domain relations, and we carefully examined the possibility of combining the different relations to establish the matching. Using domain-specific characteristics of the background knowledge reduces the applicability of the heuristics in the study to the scope of the domain, but still provides general insights about what can be expected when such domain-specific knowledge is applied in the matching process. Major findings of our study presented in this chapter are: first, domain-specific relations can be successfully applied in the matching process, and second, it is the combination of different relations in the background knowledge that results in the maximal benefit for the matching.

In Section 6.2 we will present a case study of using a comprehensive medical ontology to mediate the matching of two medical ontologies, then in Section 6.3 we will present a case study of using a music ontology to mediate the process of matching user's preferences with classes from music content provider's schemes, and with Section 6.4 we will conclude the chapter.

## 6.2 Combining relations in medical background knowledge

Aiming to analyze the effects of combining relations in the background knowledge, in the first case study we conducted a set of experiments with medical ontologies. In choosing a representative test data for experimenting, we used the following selection criteria: *(i)* the matching ontologies should cover the same domain, *(ii)* the background knowledge should be of a high quality (preferably created by specialists in the field), and *(iii)* besides covering the same domain as the matching ontologies it should be comprehensive and rich with different relations. Matching these criteria, we choose three ontologies: the anatomy parts of CRISP and MeSH document classification schemes as matching ontologies, and the FMA anatomy ontology as a background knowledge. As a matching task we pose the challenge to find *equivalent*, *broader-than* and *narrower-than* relations between the matching concepts. As a use-case we set to use the matches for document reclassification from CRISP to MeSH. This matching task is realistic and natural in the context, given that these schemes are used in document classification systems.

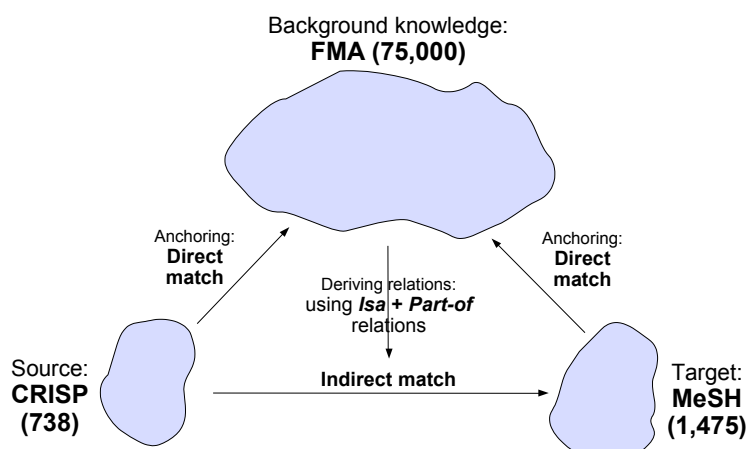


Figure 6.1: Matching CRISP to MeSH ontology using FMA as background knowledge.

## 6.2.1 Case study description

We compared the results of matching CRISP to MeSH directly, and matching them indirectly through FMA as a background knowledge. Particularly interesting are the matches found in the indirect matching, which were not found in the direct matching<sup>4</sup>. The quality of these matches, and the difference in the amount of direct and indirect matches provide a clue about the added value of the domain background knowledge in the matching.

**Source ontology:** CRISP<sup>5</sup> (Computer Retrieval of Information on Scientific Projects<sup>6</sup>) is a biomedical document classification system. The documents are classified using a scheme of medical descriptions organized in a hierarchy. The hierarchical ordering is based on *broader-than* ( $\succeq$ ) (and its inverse *narrower-than* ( $\preceq$ )) relations between the meaning of these descriptions. We extracted the anatomy sub hierarchy of this scheme and used it as a source ontology. It consists of 738 descriptions.

**Target ontology:** MeSH (Medical Subject Headings) is the National Library of Medicine's controlled vocabulary thesaurus<sup>7</sup> used for classification

<sup>4</sup>There were matches discovered in the direct and not in the indirect matching. We will discuss this, and show an example of such a match later in Section 6.2.3.

<sup>5</sup>Besides CRISP, we will also use CSP for reference, and for MeSH we will also use MSH.

<sup>6</sup><http://crisp.cit.nih.gov/>

<sup>7</sup><http://www.nlm.nih.gov/mesh/>

of documents<sup>8</sup>. For our experiments we used the anatomy sub hierarchy, which consisted of 1475 classes. As CRISP, the hierarchy of MeSH is also established through *broader-than* and *narrower-than* relations.

**Background knowledge ontology:** FMA (Foundational Model of Anatomy) as stated in its description<sup>9</sup>:

”The Foundational Model of Anatomy is a domain ontology that represents a coherent body of explicit declarative knowledge about human anatomy.”

We used a version of FMA which dates from the end of 2005. It contains 75,000 concepts interconnected with around 160 different relation types. These relation types include hierarchical orderings based on *isa* and *part-of* relations, and also include other anatomy-specific relation types like *is-attached-to*, *sends-output-to*, etc. Given our matching task, i.e. finding *broader-than* and *narrower-than* relations between the matching concepts, from the available relation types in FMA we selected the *isa* and *part-of* including the three specializations of the *part-of* relation type: *regional-part-of*, *systemic-part-of* and *constitutional-part-of*. They were the only relation types compatible with the expected matching result (*isa* and *part-of* are a special kind of *narrower-than* and consequently could be used to deduce potential indirect matches).

We generalized the three *part-of* specializations, and merged them with the existing *part-of* relations in FMA. As stated in the seminal paper on the structure of FMA [Rose and Jr., 2003], merging these specializations with their more general counter-part is valid, and results in a richer general *part-of* hierarchy. This generalization resulted in one effective *part-of* hierarchy of around 22,000 relations. As such we used it in the experiments. The subsumption hierarchy we used unchanged, as it exists in FMA. It contained all the 75,000 concepts from the ontology.

## 6.2.2 Experiments

We performed five experiments. In the first (Experiment 1), we matched CRISP to MeSH directly, and in the other four we matched them indirectly using FMA as a background knowledge. In these four experiments we tested

<sup>8</sup>We used the Dutch translation of MeSH in a study described in the previous chapter

<sup>9</sup><http://sig.biostr.washington.edu/projects/fm/AboutFM.html>

```

    The set of direct matches is empty in the beginning
1  dmatches :=  $\emptyset$ 

    Lexical phase: find equivalent and partial lexical matches
2  for every concept pair  $X \in \mathcal{C}^{\text{CSP}}, Y \in \mathcal{C}^{\text{MSH}}$  do
3    if FULLLEXMATCH( $X, Y$ ) then
4      dmatches  $\leftarrow (X \stackrel{\equiv}{\rightarrow} Y)$ 
5    if PARTIALLEXMATCH( $X, Y$ ) then
6      dmatches  $\leftarrow (X \stackrel{\prec}{\rightarrow} Y)$ 
7    if PARTIALLEXMATCH( $Y, X$ ) then
8      dmatches  $\leftarrow (X \stackrel{\succ}{\rightarrow} Y)$ 
9  end for

    Structural phase: use the structure to find more matches
10 for every two relations  $(X_1 \preceq X_2) \in \mathcal{R}^{\text{CSP}}, (Y_1 \preceq Y_2) \in \mathcal{R}^{\text{MSH}}$ 
11   if  $(X_2 \stackrel{\prec}{\rightarrow} Y_1) \in \text{dmatches}$  then
12     dmatches  $\leftarrow (X_1 \stackrel{\prec}{\rightarrow} Y_2)$ 
13   for every two relations  $(X_1 \succeq X_2) \in \mathcal{R}^{\text{CSP}}, (Y_1 \succeq Y_2) \in \mathcal{R}^{\text{MSH}}$ 
14     if  $(X_2 \stackrel{\succ}{\rightarrow} Y_1) \in \text{dmatches}$  then
15       dmatches  $\leftarrow (X_1 \stackrel{\succ}{\rightarrow} Y_2)$ 

```

Figure 6.2: Algorithm for matching CRISP to MeSH ontology directly.

different ways of using *isa* and *part-of* relations from FMA to obtain *broader-than* and *narrower-than* relations between the matching concepts. The result of each experiment was a set of matches connected through one of the three relations:  $\stackrel{\equiv}{\rightarrow}$  (*equivalent*),  $\stackrel{\prec}{\rightarrow}$  (*narrower-than*),  $\stackrel{\succ}{\rightarrow}$  (*broader-than*).

## Direct matching

**Experiment 1:** In the *direct matching* experiment we used the concept's labels and the hierarchies of the matching ontologies. It proceeded in two consequent phases: lexical and structural.

The lexical phase discovered full and partial lexical matches between the matching concepts. Before matching, in a preprocessing step we normalized the labels by discarding interpunction and stop-words like *the*, *of*, *and* etc. When comparing the labels we accounted for a different word order. When comparing separate words from the labels we considered singular/plural forms of the same words - two words have the same meaning if one differs only in an additional *s* at the end, or one word ends with *y* and the other with *ies*. This lexical matching is similar to the one we used in

## 6.2. COMBINING RELATIONS IN MEDICAL BACKGROUND KNOWLEDGE101

4.3.1, but now it is adjusted to English, while previously it was adjusted to the Dutch language.

We concluded a full match between two arbitrary concepts  $X$  and  $Y$ , i.e.  $X \equiv Y$  when a pair of their labels matched, i.e. both labels consisted of the same words. When matching partially, for two comparing arbitrary concepts  $X$  and  $Y$ , if  $X$  had a label consisting of a superset of words of a label of  $Y$  we concluded  $X \supseteq Y$ , and analogously  $X \supseteq Y$  if  $X$  had a label that consisted of a subset of the words of a label of  $Y$ . In other words, we used the partial lexical matches following the intuition that additional words in a label additionally constrain the meaning. This way, for example, we concluded that

$$\text{CSP: Mesenteric artery} \supseteq \text{MSH: Artery}$$

because *Mesenteric artery* has one extra word.

In the structural phase of direct matching we used the structure of CRISP and MeSH to further induce matches by combining the structural relations from CRISP and MeSH with the previously obtained lexical matches. For example, combining the two relations:

$$\begin{array}{l} \text{CSP: Brain} \equiv \text{MSH: Brain} \\ \text{MSH: Brain} \supseteq \text{MSH: Temporal lobes} \end{array}$$

can imply the match:

$$\text{CSP: Brain} \supseteq \text{MSH: Temporal lobes}$$

To exhaustively use the structure of the matching ontologies, we performed a transitive chaining over the combined lexical matches and structural relations from the matching ontologies. We achieved this by applying the following rules:

$$\begin{array}{l} \text{if } (X^C \supseteq Y^M) \text{ and } (Y^M \preceq Z^M) \quad \text{induce } (X^C \supseteq Z^M) \\ \text{if } (X^C \preceq Y^C) \text{ and } (Y^C \supseteq Z^M) \quad \text{induce } (X^C \supseteq Z^M) \\ \text{if } (X^C \supseteq Y^M) \text{ and } (Y^M \succeq Z^M) \quad \text{induce } (X^C \supseteq Z^M) \\ \text{if } (X^C \succeq Y^C) \text{ and } (Y^C \supseteq Z^M) \quad \text{induce } (X^C \supseteq Z^M) \end{array}$$

Even though equivalence ( $\equiv$ ) relations are not found in the rules, they were implicitly used. Namely, we considered  $X \equiv Y$  as the two relations  $X \preceq Y$  and  $X \succeq Y$  compounded, and used them as such in the rules above. The final result of the direct matching was obtained after exhaustive application of these rules on the set of matches obtained in the lexical phase. The complete algorithm for directly matching the CRISP to MeSH is shown in Figure 6.2.

```

    The set of indirect matches is empty in the beginning
1  imatches :=  $\emptyset$ 

    Anchoring phase: anchor CRISP and MeSH to FMA using direct matching
2  anchC→F := MATCHDIRECTLY(CRISP, FMA)
3  anchM→F := MATCHDIRECTLY(MeSH, FMA)

    Deriving relations phase: find indirect matches using the anchors and FMA
4  for every two anchors  $(X \mapsto Z_1) \in \text{anch}^{C \mapsto F}$ ,  $(Y \mapsto Z_2) \in \text{anch}^{M \mapsto F}$ 
5    if  $(Z_1 \preceq Z_2)$  then
6      imatches  $\leftarrow (X \mapsto Y)$ 
7  for every two anchors  $(X \mapsto Z_1) \in \text{anch}^{C \mapsto F}$ ,  $(Y \mapsto Z_2) \in \text{anch}^{M \mapsto F}$ 
8    if  $(Z_1 \succeq Z_2)$  then
9      imatches  $\leftarrow (X \mapsto Y)$ 

```

Figure 6.3: Algorithm for matching CRISP to MeSH indirectly through FMA as a background knowledge. MATCHDIRECTLY is depicted in Figure 6.2.

### Indirect matching

In the indirect matching experiments we applied the general matching scheme shown in Figure 3.1 from Chapter 3. As such, the matching proceeded in two steps: anchoring CRISP and MeSH to FMA, and then finding indirect matches between CRISP and MeSH through FMA, using it as a background knowledge (deriving relations step). The general matching scheme instantiated for this particular case is shown in Figure 6.1.

We anchored CRISP and MeSH to FMA using the lexical matching technique described in Experiment 1 to match CRISP to MeSH. Both CRISP and MeSH were anchored to FMA, producing two result sets:  $\text{CSP} \mapsto \text{FMA}$  and  $\text{MSH} \mapsto \text{FMA}$  with anchor relations of the three types:  $\mapsto^{\equiv}$ ,  $\mapsto^{\preceq}$  and  $\mapsto^{\succeq}$ . The produced anchors were used in all four indirect matching experiments (Exp.2 - Exp.5).

When deriving the relations, we combined the anchors with the relations in the background knowledge. Basically, we chained the  $\preceq$  and  $\succeq$  relations across the anchors and the background knowledge<sup>10</sup>. We did this by applying the two symmetric rules:

$$\begin{array}{ll}
 \text{if } (X^C \mapsto^{\preceq} Z_1^F \preceq Z_2^F \mapsto^{\preceq} Y^M) & \text{induce } (X^C \mapsto^{\preceq} Y^M) \\
 \text{if } (X^C \mapsto^{\succeq} Z_1^F \succeq Z_2^F \mapsto^{\succeq} Y^M) & \text{induce } (X^C \mapsto^{\succeq} Y^M)
 \end{array}$$

<sup>10</sup>We did something similar in the direct matching where we chained the  $\succeq$  and  $\preceq$  relations obtained in the lexical matching with the structure of the ontologies

These rules were applied exhaustively, and the result was the indirect matching between CRISP and MeSH. Note that these rules use equivalence relations as a special type of  $\succeq$  and  $\preceq$ , and the anchors established from MeSH are in the reverse direction, i.e. they anchor MeSH concepts to the FMA concepts. The complete algorithm that performs the described indirect matching is shown in Figure 6.3.

The rules above use  $\preceq$  and  $\succeq$  relations between the concepts in FMA ( $Z_1^F \preceq Z_2^F$  and  $Z_1^F \succeq Z_2^F$ ). However, as explained in its description, FMA does not contain any relations of this type. Instead, it contains its specializations: *isa*, *part-of*, and three special types of *part-of* relations. All the indirect matching experiments generalized these relations to  $\preceq$  and  $\succeq$ , but they did it in different ways. Now, we describe how each indirect matching experiment used the *isa* and *part-of* relations from FMA.

**Experiment 2:** *Indirect matching by using FMA isa and part-of direct relations.* We induced a relation between the FMA concepts if they were directly related with *isa* or *part-of* relation. We used the following rules:

$$\begin{array}{ll} (X^F \text{ isa } Y^F) & \text{induce } (X^F \preceq Y^F) \\ (X^F \text{ part-of } Y^F) & \text{induce } (X^F \preceq Y^F) \end{array}$$

When a relation  $X^F \preceq Y^F$  was induced, we added its semantic equivalent  $Y^F \succeq X^F$  as well. We did this symmetric inference in all the indirect matching experiments.

**Experiment 3:** *Indirect matching by using the transitive chaining of FMA isa and part-of relations independantly.* Relation between two FMA concepts was induced when they were related in a *isa\** or *part-of\** relation, that means using *isa* or *part-of* multiple times. This was achieved by applying the following rules:

$$\begin{array}{ll} (X_1^F \text{ isa } \dots \text{ isa } X_n^F) & \text{induce } (X_1^F \preceq X_n^F) \\ (X_1^F \text{ part-of } \dots \text{ part-of } X_n^F) & \text{induce } (X_1^F \preceq X_n^F) \end{array}$$

**Experiment 4:** *Indirect matching by using the transitive chaining of FMA isa and part-of relations combined.* In this experiment we merged the *isa* and *part-of* relations and then used the transitive chaining of the resulting relation. We used one single inference rule:

$$(X_1^F \sim^1 \dots \sim^n X_n^F) \text{ where } \sim^i \in \{\text{isa}, \text{part-of}\} \text{ induce } (X_1^F \preceq X_n^F)$$

After analyzing the results of Experiment 4 it appeared that wrong inferences were created when *isa* relation is used before *part-of* in the chaining, later in this section we will see an example of this. To overcome this negative effect we did the next experiment.



**Experiment 5:** *Indirect matching by using the transitive chaining of FMA isa and part-of relations combined, without using isa before part-of.* What we did in this experiment was using the same inferencing as in the previous experiment but we avoided the use of *isa* relation before *part-of* in the chaining process. Such inference was achieved using the following rule:

$$(X_1^{\text{F}} \text{ part-of} \dots \text{ part-of } X_k^{\text{F}} \text{ isa} \dots \text{ isa } X_n^{\text{F}}) \text{ induce } (X_1^{\text{F}} \preceq X_n^{\text{F}})$$

### 6.2.3 Analysis of the results

Now we will present, evaluate and analyze the results of the experiments. Before doing so, we will first explain how we will characterize and present the matching results.

An important issue in presenting the matching results is that a set of matches may contain matches which are implied by the others, in combination with the structure of the ontologies. For example, all the concepts in CRISP are found more specific than the root concept in MeSH, whereas having equivalence between the two root concepts already implies all those matches. Similarly, having a match between two concepts contains implicit knowledge about their sub and super-concepts. To make a fair trade-off between the two cases of having all the possible matches and having only the minimal set of matches that implies all the rest, we decided on a result set that is in between.

In each matching experiment we did the following: We started from the set of all matches, including the implied. For each source concept we took the set of all its matches, and then minimized that set by discarding the matches which are implied by the rest of the set. The minimal set is not sensitive to the order of discarding the implied matches. The union of these minimized sets was the final result. This trade-off matching set extracts the minimal knowledge from the matching result for each of the source concepts separately.

In the anchoring phase we matched CRISP and MeSH to FMA directly. The results are shown in Figure 6.4. The equivalence relations were established as 1-1 matches, while *narrower-than* and *broader-than* as many to many. Looking for equivalences only already produced successful anchoring: 65.5% of CRISP and 70.6% of MeSH concepts were anchored to their equivalent concepts in FMA. This success comes from the richness of FMA. Yet, for many there were no equivalent concepts in FMA because of different scopes of coverage of the anatomy domain. In CRISP, for example, there is a concept CSP: Muscle movement which is not an anatomical part of the human

body, and as such does not exist in FMA. Still, as shown in the last column on Figure 6.4, nearly 99% of the concepts from both CRISP and MeSH were anchored due to the use of the structure of CRISP and MeSH. For example, CSP:Muscle movement was anchored as *narrower-than* to FMA:Muscle because within CRISP it is *narrower-than* CSP:Muscle.

	Anchoring concepts	$\equiv$	$\prec$	$\succ$	Anchored concepts
CRISP $\mapsto$ FMA	738	483 (65%)	607	1,474	730 (98.9%)
MeSH $\mapsto$ FMA	1,475	1,042 (70%)	1,545	2,227	1,462 (99.1%)

Figure 6.4: The number of anchors obtained when anchoring CRISP and MeSH to FMA

Figure 6.5 summarizes the results of the five experiments. Comparing the indirect to the direct matching, the indirect matchings found many more *narrower-than* and *broader-than* relations than the direct matching. It appeared that the concepts in CRISP and MeSH can be related in many more ways beyond those found by using the structure of these ontologies alone. In this case FMA contributed the missing knowledge which resulted in the improvement over the direct matching.

Matches CRISP $\rightarrow$ MeSH	$\equiv$	$\prec$	$\succ$	Total	Increase
Exper. 1: Direct	448	417	156	1,021	
Exper. 2: Indir. <i>isa</i> , <i>part-of</i>	395	516	405	1,316	29%
Exper. 3: Indir. <i>isa*</i> , <i>part-of*</i>	395	933	1,402	2,730	167%
Exper. 4: Indir. ( <i>isa</i> , <i>part-of</i> )*	395	1,511	2,228	4,143	306%
Exper. 5: Indir. <i>part-of*</i> <i>isa*</i>	395	972	1,800	3,167	210%

Figure 6.5: The number of matches in the five experiments of matching CRISP to MeSH directly and indirectly

The last column in Figure 6.5 shows the increase in the number of matches obtained in the indirect matching relative to the direct matching. The indirect matching of Experiment 2 produced 29% more matches than the direct matching. So, using only the direct *isa* and *part-of* relations between the concepts in FMA already outperformed the direct matching. When using the transitive closure of *isa* and *part-of* (Experiment 3) we obtained an increase of 167%, or nearly 2.7 times more matches than the direct matching. When arbitrarily mixing *isa* and *part-of* with their transitive closure we got an

increase of 306%, or 4 times more matches than the direct. The fifth experiment, when combining the *isa* and *part-of* in a restricted way, there was an increase of 210% which is 3.1 times more matches than the direct. It produced 26% less matches than the fourth, and 19% more than the third experiment. We observe that the first major jump in the performance is noticeable when the transitive chaining is involved in the inference, that is in Experiment 3 (167% increase).

These numbers show that using background knowledge produces substantially more matches than direct matching. Without combining the relations within the background knowledge it is already better than the direct matching, then combining the relations in the background knowledge produces much more matches, and combining different relations within the background knowledge produces the maximal benefit of this approach. Of course, these numbers do not say anything about the quality of the matches. In particular, if the relations are combined arbitrarily then there is big increase in the number of matches but also wrong matches are created as we will see below. However, when combining the background relations in a specific way we retain the precision while still considerably increasing the recall.

For the *Equivalence* ( $\equiv$ ) relation, the indirect matching found slightly less relations than the direct matching. All the indirect matchings discovered the same amount of equivalences because the only way to find equivalence indirectly is to have both concepts anchored as equivalent to the same concept in the background knowledge (this is not the case in general, if the background knowledge contains loops the reasoning can find equivalent concepts even when they are not anchored to the same background concept. This occurs in practice, however, it is a problem of the modeling of the background knowledge, not the matching method). The equivalences found directly and not indirectly were caused by concepts which existed in CRISP and MeSH but not in FMA. In the next section we discuss such a case. In few cases equivalences were detected indirectly and not directly because their labels were found as synonymous only through the background knowledge.

**Evaluation of results** To test for correctness of the matches, we randomly choose 70 CRISP concepts, and inspected their matches found in the experiments by manually browsing the Wikipedia<sup>11</sup> pages describing these concepts. The sample set is about 10% of the size of CRISP. The results of the evaluation are presented in Figure 6.6. The table describes the evaluation for each experiment in a single row: column two is the number of

---

<sup>11</sup><http://wikipedia.org/>

## 6.2. COMBINING RELATIONS IN MEDICAL BACKGROUND KNOWLEDGE<sup>107</sup>

obtained matches, column 3 is the number of correct matches, column 4 is the change in recall as compared to the direct matching, and the column 5 is the precision measured in the experiment.

Matches CRISP → MeSH	Total	Correct	$\Delta$ Recall	Precision
Exp.1: Direct	80	80	/	100%
Exp.2: Indir. <i>isa</i> and <i>part-of</i>	107	107	+34%	100%
Exp.3: Indir. <i>isa part-of</i>	193	193	+141%	100%
Exp.4: Indir. ( <i>isa, part-of</i> )*	303	274	+245%	90.4%
Exp.5: Indir. <i>part-of*</i> <i>isa*</i>	228	228	+185%	100%

Figure 6.6: Evaluation of the matchings CRISP to MeSH directly and indirectly on a randomly drawn set of 70 CRISP concepts

Inspecting the precision, only in Experiment 4 there were wrong matches found where the precision dropped to 90.4%. In the other experiments it was 100%, meaning that all the discovered matches were correct. The high percentage of correct matches in the indirect matching experiments can be explained with the high quality of the knowledge in FMA. However, in lack of gold standard, checking the correctness of the matched turned out not to be straight forward. Often, we had to make a non-trivial choice what to consider correct and what not. Namely, some matches are arguably correct because of the non-precise definition of the *broader-than* (and *narrower-than*) relation type. When using the relation *constitutional-part-of* from FMA, for example, we found that the ulnar artery is a constitutional part of the elbow, but it stretches through the whole arm, and therefore it is not part of the elbow only. We call these matches shared. Having the relation between ulnar artery and elbow is a useful one, somebody looking for medical resources about an elbow is interested in the arteries passing through the elbow as well, see Wikipedia for more details on this example. We explored the matches  $\preceq$  or  $\succeq$  produced in Experiment 5 on the Gold Standard set of concepts (in total they were 185 matches), and found out that 60 matches are shared, while the other 125 are not. This means that even if inspecting the matches rigidly by discarding shared matches, the background knowledge still results in a large boost of the matching results.

The *recall* was too difficult to reliably estimate for the matching results<sup>12</sup>. In lack of a Gold Standard to calculate the recall, we observed its relative change over the different experiments. This change is shown in the table in

<sup>12</sup>To estimate the recall on a given sample set, we should be able to find all the possible correct matches for each concept in the set.

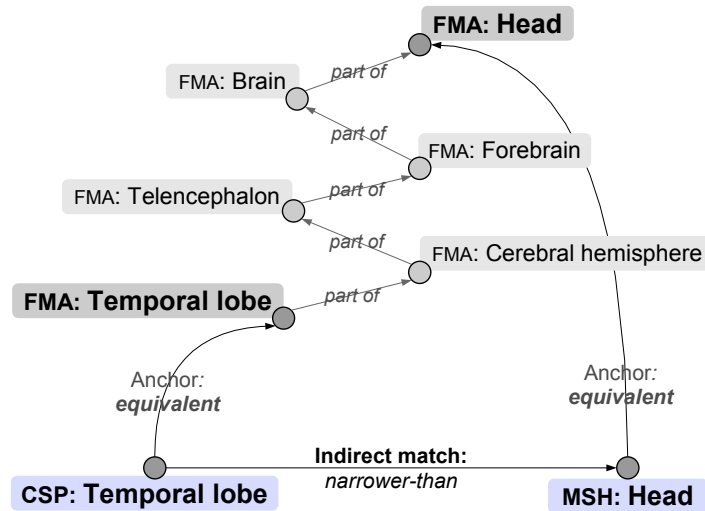


Figure 6.7: Example of a correct indirect match  $\text{Temporal lobe} \xrightarrow{\preceq} \text{Head}$ .

Figure 6.6 in the fourth column. The highest increase in recall was achieved in Experiment 4, which resulted in an increase of 245% as compared to the direct matching.

**Example matches** To get a clearer picture of the matching process, we discuss three example matches which are representative for different classes from the result sets.

- *Case 1: Matches found by indirect and not by the direct matching.* A representative example of these matches is:

$$\text{CSP: Temporal lobe} \xrightarrow{\preceq} \text{MSH: Head}$$

Its inference is shown in Figure 6.7. Temporal lobes are parts of the brain, and consequently parts of the human's head. In the structure of MeSH and CRISP they are classified under the Brain which is classified under Central Nervous System, and are not connected in any way with the Head. That prevents straight forward detection of the match in the direct matching, simply using the relations in the matching ontologies. As a consequence the background knowledge is crucial in finding the match.

- *Case 2: Incorrect match produced by arbitrary mixing of isa and part-of hierarchies.* When using *isa* and then *part-of* in the inference, some of the matches were incorrect. An example of such an inference is shown in Figure 6.8, finding the match:

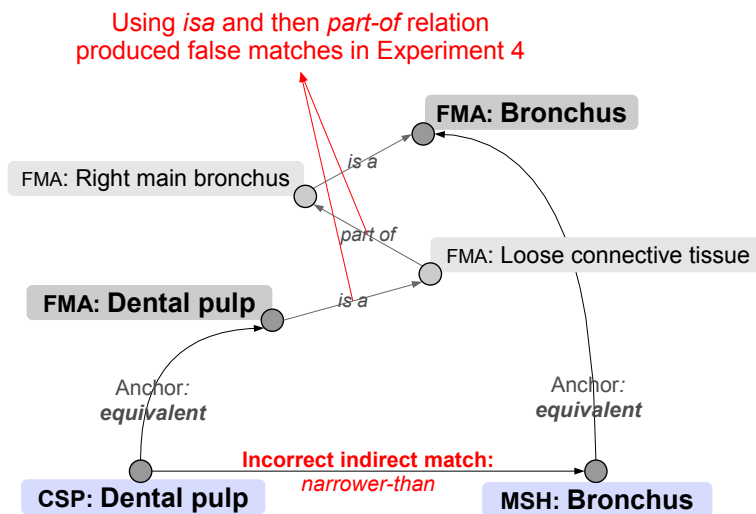


Figure 6.8: Example of an incorrect indirect match in Experiment 4: Dental pulp  $\xrightarrow{\preceq}$  Bronchus.

$$\text{CSP: Dental pulp} \xrightarrow{\preceq} \text{MSH: Bronchus}$$

Dental pulp is a tissue located in the center of a tooth, and bronchus is an air passage that leads from the trachea to the lung. Obviously, the two can not be connected in a *broader-than* relation. So, what causes this wrong inference? Each of the two relations:

$$\begin{array}{lcl} \text{FMA: Dental pulp} & \textit{isa} & \text{FMA: Loose conn. tissue} \\ \text{FMA: Loose conn. tissue} & \textit{part-of} & \text{FMA: Right main bronchus} \end{array}$$

remain correct when *isa* and *part-of* are generalized to  $\preceq$ :

$$\begin{array}{lcl} \text{FMA: Dental pulp} & \preceq & \text{FMA: Loose conn. tissue} \\ \text{FMA: Loose conn. tissue} & \preceq & \text{FMA: Right main bronchus} \end{array}$$

however, the the chaining over this generalization is not valid any more, the relation:

$$\text{Dental pulp} \preceq \text{Right main bronchus}$$

is incorrect, which then leads to the incorrect match:

$$\text{CSP: Dental pulp} \xrightarrow{\preceq} \text{MSH: Bronchus}$$

- *Case 3: Matches found by direct but not by the indirect matching.* An example of such a match is

$$\text{CSP: Mesenteric artery} \xrightarrow{\equiv} \text{MSH: Mesenteric Arteries}$$

The relation was not found indirectly because a concept *Mesenteric artery* does not exist in FMA. Instead, in FMA there are two more

specific FMA: Inferior Mesenteric Artery and FMA: Superior Mesenteric Artery concepts, and one broader FMA: Artery. Using FMA the following indirect matches were found for CSP: Mesenteric artery:

CSP: Mesenteric artery	$\xrightarrow{\text{is a}}$	MSH: Arteries
CSP: Mesenteric artery	$\xrightarrow{\text{is a}}$	MSH: Mesenteric Artery, Inferior
CSP: Mesenteric artery	$\xrightarrow{\text{is a}}$	MSH: Mesenteric Artery, Superior

As can be observed from this example, combining the direct and indirect matching would improve the results of the both.

#### 6.2.4 Lessons learnt

Based on the conducted experiments, we conclude that using a domain comprehensive background ontology can boost the ontology matching process as compared to a direct matching of the two ontologies. Most of the value in using the background knowledge comes from combining the relations within the background knowledge. Transitive chaining of the relations in the background knowledge provided the largest jump in the matching performance. Finally, the different relations need a careful combination in order to gain maximal benefit, while new ways to combine relations result in additional matches, some combinations may result in wrong matches as well.

A crucial requirement in using background knowledge is the existence of extensive reference ontologies in different domains at hand. Therefore, the development of such ontologies and subsequent publication on the Semantic Web, or other means that would make them available, will make the problem of integration easier.

Our findings were concluded from experiments conducted on medical test data. Next, we will present a scheme of using domain background knowledge in the music domain. In contrast to the medical domain, the music domain suffers from no standardized vocabulary for communication. The people are confronted with the problem to find the right search terms in order to find the content of interest. In this direction, using background knowledge can help to bridge the gap between the user's and the content provider's vocabulary.

## 6.3 Boost music search using domain background knowledge

In this section we will describe a case study in which we applied a music domain ontology as a background knowledge in an ontology matching problem. In line with the topic of this chapter, we focus on combining different relations in the background ontology.

Our problem in this case is matching user's preferences with the music provider classifications schemes. The music content offered by the music providers' portals is annotated with a metadata which is used to organize the offered content. Now, our matching problem is to find a correspondence between the user's preferences and the metadata of the provider's classification schemes.

Various music resources are available on the Internet. A couple of famous examples include MusicMoz<sup>13</sup>, MusicBrainz<sup>14</sup> and AllMusic<sup>15</sup>. Their content is organized in classes of music like genres and styles, which mainly contain artists, releases and songs. These entities are further described through additional metadata about the origin of the music, year of release, record label, etc. Rich descriptions are increasingly becoming available: Semantic Wikipedia<sup>16</sup> is an attempt to formalize the content of Wikipedia (or parts of it) in the form of an ontology, AllMusic<sup>17</sup> contains very rich metadata which is implicitly offered on the web site as an ingredient that would help the users find their way on the portal. MusicBrainz<sup>18</sup> is a freely available large collection of metadata of artists, releases and songs. Their metadata is currently limited to an objective (technical) metadata such as year of release, origin etc. but lacks classification in genres and other descriptive metadata which is based on a rather subjective criteria.

The user's preferences when expressed in one of the simplest forms, can be a music search query expressed as a string consisting of several words<sup>19</sup>. Then the problem becomes matching the search query with the most relevant target classes. As discussed in [Van Rijsbergen, 1979] finding which target classes are relevant to the search query is a subjective problem. Usually the

---

<sup>13</sup><http://www.musicmoz.org/>

<sup>14</sup><http://www.musicbrainz.org/>

<sup>15</sup><http://www.allmusic.com/>

<sup>16</sup>[http://wiki.ontoworld.org/wiki/Main\\_Page](http://wiki.ontoworld.org/wiki/Main_Page)

<sup>17</sup><http://allmusic.com/>

<sup>18</sup><http://musicbrainz.org>

<sup>19</sup>This is a form of information retrieval problem [Baeza-Yates et al., 1999, Van Rijsbergen, 1979, Liu et al., 2004]



system returns multiple answers which are ranked according to an estimated relevancy.

When a user is searching for music, he is faced with two main problems: find the right vocabulary that he needs to use in order to properly describe his preferences as they would match the metadata in the system, and find the music providers on the Internet. Having no central point for searching, he has to search through each content provider's site separately. We will now describe a search scheme that provides a solution to both of these problems: the users can search using a richer vocabulary than the one used in the providers' metadata, and they have a unified access to the different providers by performing one single search. The search scheme involves the use of a background knowledge ontology, and as we will see through examples, some typical searches require the background knowledge for a successful retrieval of the content of interest.

Based on that search scheme we built a prototype music search system. It gives the possibility to search through music classes from providers schemes from existing Internet music providers, while using background knowledge. The background knowledge ontology was manually constructed by reading and analysing the music pages in Wikipedia. In the next sections, we will first give an overview of the landscape of music provision on the Internet, then we will describe the scheme of searching music which was applied in the prototype system, then we will provide a limited evaluation of the prototype, to finally discuss this proposed solution.

### 6.3.1 Music domain on the Internet

When formalizing the knowledge contained in the music metadata, one immediately encounters a serious problem of the domain. The domain suffers from two intrinsic problems concerning the definition of a music genre. First, music genre is intrinsically ill-defined, as discussed in [Pachet and Cazaly, 2000, Aucouturier and Pachet, 2003]. Two ways to define it are most widely accepted: extensional - genre is represented by a set of rules that a piece of music has to comply with in order to be part of that genre; and intentional - genre is simply a set of music entities, and a piece of music belongs to the genre when it belongs to its corresponding set. Second, there is no ground truth or a central authority to decide how different music entities should be assigned to genres or styles<sup>20</sup>. In an earlier study we showed that even for a general concept like *Rock* the famous portals like MusicMoz and ADN

---

<sup>20</sup>This also includes other metadata such as mood, tempo, etc.

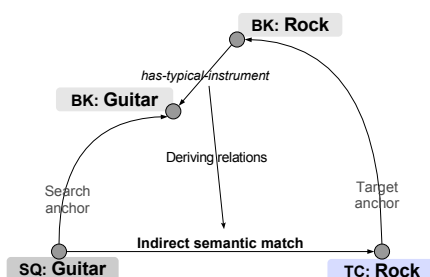


Figure 6.9: Example of indirect match in the context of music.

disagree to a high degree on the set of artists that constitutes the genre, when only looking at the shared artists which are classified in both of them [Aleksovski et al., 2004]. This fact is also reflected in Wikipedia; for some of the music styles, like for example *Trance*, quite active discussion is ongoing about its precise description and characterisation.

### 6.3.2 Music test data extracted from Internet

To simulate a realistic music search scenario, we selected and extracted the provider's schemes of 7 online music providers to serve as a target, and in lack of domain ontology we manually constructed one ourselves using Wikipedia as a source of music knowledge.

**Target ontologies: Seven classifications of music providers** Most of the providers offer the music classified in genres and styles for an easy user browsing. Visitor to the site can interactively navigate through different pages that list the music offered. There is an implicit underlying structure of navigation paths together with the labeling on the links and pages. After considering several of the online music sites, we selected seven of them and extracted the schema underlying the classification of music they offer, through the navigation paths. They were: Amazon<sup>21</sup> (CdNow), MusicMoz<sup>22</sup>, CD Baby<sup>23</sup>, Artist Direct Network<sup>24</sup> (also called ADN), AllMusic<sup>25</sup>, Launch

<sup>21</sup><http://amazon.com/>

<sup>22</sup><http://musicmoz.org/>

<sup>23</sup><http://www.cdbaby.com/>

<sup>24</sup><http://www.artistdirect.com/>

<sup>25</sup><http://allmusic.com/>

<b>CDNOW (Amazon.com)</b>	Size <b>2410</b> classes Depth <b>5</b> levels	
<b>MusicMoz</b>	Size <b>1073</b> classes Depth <b>7</b> levels	
<b>Artist Direct Network</b>	Size <b>465</b> classes Depth <b>2</b> levels	
<b>All Music Guide</b>	Size <b>403</b> classes Depth <b>3</b> levels	
<b>Artist Gigs</b>	Size <b>382</b> classes Depth <b>4</b> levels	
<b>CD baby</b>	Size <b>222</b> classes Depth <b>2</b> levels	
<b>Yahoo LaunchCast</b>	Size <b>96</b> classes Depth <b>2</b> levels	

Figure 6.10: The sizes and depths of the seven ontologies used as target in the system.

cast on Yahoo<sup>26</sup> and ArtistGigs<sup>27</sup>. Their sizes and depths are shown in Figure 6.10.

The extracted classifications we transformed into ontologies, each consisting of a single hierarchy of concepts. Each concept corresponds to a class in the classification. The concept is named with a label, and possibly with synonyms to that label, if they existed on the source site.

Commercial providers often include classes whose meaning lies outside the music styles domain, for example *Music Accessories*. These are excluded from the experiments. Each schema has its own peculiarities that are typical or unique to it, Amazon contains classic composers as separate classes, ADN mainly divides the music by the period of issuing/publication, etc.

**Background knowledge - MusicOracle, a manually constructed ontology using Wikipedia** In need for a rich ontology describing the music domain, after extensive search on the Internet, we decided to build one ourselves by manually extracting it from the online encyclopedia - Wikipedia. This resource offers a description of music genres and styles, organized in web pages. Each genre or style is described in human readable form. After analyzing these pages, we concluded that it is not straightforward (if possible) to extract automatically the content we need. Manually reading and analyzing the pages seems to be required and we decided to go through the effort.

The ontology we constructed in this way we call **MusicOracle**. It is organized in four main hierarchies: *genres*, *instruments*, *time periods* and *geo-*

<sup>26</sup><http://yahoo.com/>

<sup>27</sup><http://artistgigs.com/>

*graphic locations*. The rest of the concepts were other music attributes which were less structured. The main component of the **MusicOracle** is the genre hierarchy. Each concept in this hierarchy is attributed with the rest of the concepts through music-specific relations. A genre can have multiple annotations (even within a single of the three other hierarchies). For example, genres can be related with *mainstream-popularity* and also with *period-of-emergence* relations to the hierarchy of time periods.

In the present form, the **MusicOracle** consists of 5,920 concepts. They include 1,929 genres or styles of music, 1,006 instruments and 998 geographic locations. The latter are subdivided into continents, countries, regions and cities. The remaining concepts describe time periods, like 1920's or Twentieth century, and other types of attributes, like **Female vocals**. The concepts are interconnected with 12,953 relations using 14 different relation types. The *broader-than* (and *narrower-than*) relation type is used to construct the hierarchies, and the other 13 are "attribute" kind of relation types.

### 6.3.3 Music search using a background ontology

Searching for music on the Internet can be regarded as an ontology matching problem. The source ontology in this case is the user's search query (an ontology of a single concept with one label), and the target is multitude of different music providers' classification schemes. Resembling the problem of ontology matching, we apply the paradigm of using background knowledge in this context. The modified matching scheme is shown in Figure 6.11.

To explain the matching process in the context of music, consider the example shown in Figure 6.9: the search query **SQ: Guitar** is anchored to a background knowledge concept **BK: Guitar**, and the target concept **TAR: Rock** is anchored to a background knowledge concept **BK: Rock**. The background knowledge reveals a relation **BK: Rock** *has-typical-instrument* **BK: Guitar**, and we derive a relation that the search query **SQ: Guitar** matches with the target concept **TAR: Rock**. The background knowledge revealed a non-trivial semantic relation; the match would not be found by standard lexical search through the target ontology<sup>28</sup>. This example illustrates a simple use of the background knowledge, and later we will see how the relations in the background knowledge can be further combined in the matching.

In accordance with this search scheme we built a prototype music search system. The target ontologies were the seven classifications extracted from

---

<sup>28</sup>In 2004 when the target classifications were extracted, none of the source sites were able to retrieve a sensible answer when queried with *Guitar*

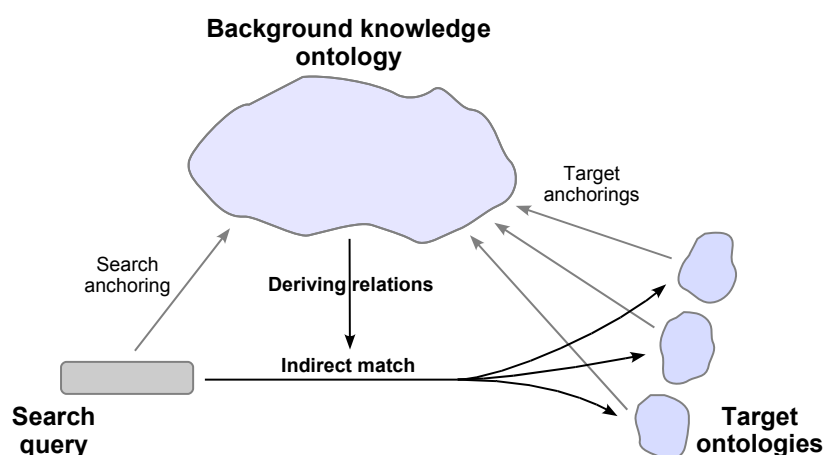


Figure 6.11: Searching through a variety of target ontologies using background knowledge.

the music sites, and the background knowledge was the MusicOracle ontology extracted from Wikipedia, see Figure 6.12. The system allows the users to search the seven music classifications at once. Based on the user's search query, which is a string consisting of several words, the system performs the search and returns a ranked list of concepts from the target ontologies. The ranking is by importance, which is derived from the strength of correspondence to the search query. We will now describe the process in detail.

In a preprocessing phase the target ontologies are anchored to the MusicOracle<sup>29</sup>. Then, each search query is anchored to the MusicOracle, and through the deriving relations step the requested results are found. The deriving relations step searches for patterns between the anchored concepts in the MusicOracle.

**The anchoring** of the search query and the target ontologies to the MusicOracle are performed lexically. The labels were normalized by discarding interpunction and general words like *the*, *of*, *and* etc., and a lightweight form of stemming was applied, accounting for word order and singular/plural forms. In the search anchoring,  $X^{\text{SQ}}$  was anchored to  $Y^{\text{MO}}$  when all the words in a label of  $Y^{\text{MO}}$  were found in the search query  $X^{\text{SQ}}$ . We write such relation

<sup>29</sup>When applied in a real system, the target classifications will change periodically, and the anchoring phase can be repeated on such a periodic basis. This phase can be seen as a sort of crawling and indexing operation

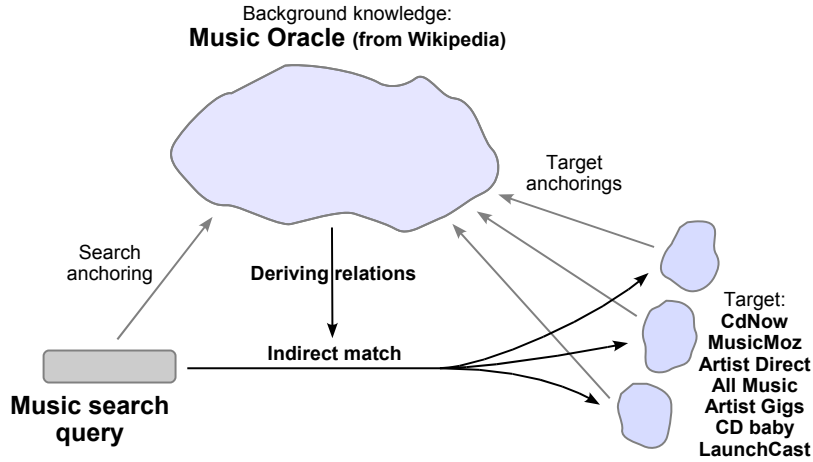


Figure 6.12: Searching through 7 music classifications using the MO ontology as background knowledge.

as  $X^{\text{SQ}} \xrightarrow{sa} Y^{\text{MO}30}$ . Analogously, in the target anchoring  $Z^{\text{TC}}$  was anchored to  $Y^{\text{MO}}$ . Such relation we write as  $Z^{\text{TC}} \xrightarrow{ta} Y^{\text{MO}}$ . The anchoring used the lexical matching described in detail in 6.2.2. Example anchorings are:

$$\begin{aligned} \text{SQ: Piano Guitar} &\xrightarrow{sa} \text{MO: Guitar} \\ \text{TC: Death metal} &\xrightarrow{ta} \text{MO: Death Metal} \end{aligned}$$

**Deriving relations.** In this step we used the music domain-specific knowledge from the background knowledge ontology. It was performed using one single generic rule, which can be expressed in the following form:

$$(X^{\text{SQ}} \xrightarrow{sa} Z_1^{\text{MO}} \sim Z_2^{\text{MO}} \xrightarrow{ta} Y^{\text{TC}}) \quad \text{induce} \quad (X^{\text{SQ}} \longrightarrow Y^{\text{TC}})$$

This rule basically says that the search query  $X^{\text{SQ}}$  is matched with the target concept  $Y^{\text{TC}}$ , when there is a search anchor from  $X^{\text{SQ}}$  to a concept  $Z_1^{\text{MO}}$  in the MusicOracle and a target anchor from  $Y^{\text{TC}}$  to a concept  $Z_2^{\text{MO}}$  in the MusicOracle, and the two concepts in the MusicOracle are related  $Z_1^{\text{MO}} \sim Z_2^{\text{MO}}$ .

The question is now which pairs of concepts from the MusicOracle should we consider related, as to the scope of the application? Besides the existing direct relations between concepts, some relations can be combined to obtain again useful pairs of related concepts. These combinations of relations we call inference patterns. Starting from the matching task, and considering

<sup>30</sup>The *sa* letters mean that the relation is a Search Anchor, *ta* means it is a Target Anchor.

Importance rank	Attribute relation types in the MusicOracle
#1	<i>related-to</i>
#2	<i>typical-instruments</i>
#3	<i>instruments</i>
#4	<i>exclusive-location</i>
#5	<i>location-of-origin</i>
#6	<i>mainstream-popularity</i>
#7	<i>location</i>
#8	<i>period-of-emergence</i>
#9	<i>other-instruments</i>
#10	<i>tempo</i>
#11	<i>lyrics</i>
#12	<i>dance</i>
#13	<i>rhythm</i>

Figure 6.13: The attribute relation types in MusicOracle ordered by importance as we used them in the inference patterns.

the current structure of the MusicOracle, after an experimental analysis we developed the following inference patterns as useful:

$(X \preceq Y)$	<b>induce</b>	$(X \sim Y)$
$(X \overset{att}{\rightsquigarrow} Y)$	<b>induce</b>	$(X \sim Y)$
$(X \overset{att}{\rightsquigarrow} Z \succeq Y)$	<b>induce</b>	$(X \sim Y)$
$(X \preceq Z \overset{att}{\rightsquigarrow} Y)$	<b>induce</b>	$(X \sim Y)$
$(X \preceq Z_1 \overset{att}{\rightsquigarrow} Z_2 \succeq Y)$	<b>induce</b>	$(X \sim Y)$
$(X \succeq Z \overset{att}{\rightsquigarrow} Y)$	<b>induce</b>	$(X \sim Y)$
$(X \succeq Z_1 \overset{att}{\rightsquigarrow} Z_2 \succeq Y)$	<b>induce</b>	$(X \sim Y)$

We applied some form of ranking on the inferred results. The relation  $\overset{att}{\rightsquigarrow}$  is generic and can be any attribute annotation in the Music Oracle. However, not all the attribute relations are of equal importance. Therefore we ranked them by importance. From the most important to the least important, the order of the relations that we used in the prototype is shown in Figure 6.13. This ranking in its current form was intuitively constructed, and can be a subject of further improvement.

After applying these inference patterns, all the possible indirect matches from the search query to the target classifications are found. Then they are ranked according to the importance of the corresponding pattern used to

establish the match, that is the importance of the appropriate attribute used in the pattern. This final list of ranked matches is shown to the user as a result to the search query.

**Preliminary evaluation of the system** Rather than creating a Gold Standard, we evaluated this prototype through a preliminary user test. Four users participated in the test. They were given two different interfaces for searching music: searching directly the seven target classifications using a pure lexical matching technique, and searching the target classifications using background knowledge. They were explained that in the second case the search interface allows the use of a richer vocabulary including terms related to music, such as instruments. They were left to use the two interfaces to search through the classifications, and then they were asked to answer the following questions:

- Do you find it useful using richer vocabulary in the search process?
- Do you agree with the ranking of the results?

The intuition behind these questions corresponds to measuring the recall in the first, and the precision in the second question. On the first question we got positive answers from all four participants, they all considered having a richer vocabulary as a nice feature in a music search system. On the second question, two participants had positive reaction on the ranking scheme, and two did not agree with it and commented that the ranking of the results can be further improved.

All the participants gave the feedback that they would rather prefer to get more precise music content as a search result. When querying they would expect artists, releases and/or songs as results, the genres and styles come lower in their preference for the retrieved results. This very same scheme can be applied to perform a search which would retrieve such results, but for that purpose we needed artists, releases and songs in the target classification, and also their rich annotations in the background knowledge. This is, however, a problem of availability of data, not the search scheme.

### 6.3.4 Discussion

In our preliminary tests of the Music Search System we used as a background knowledge an ontology extracted from Wikipedia, and, according to



the user tests it produced promising results. The described prototype can be improved in multiple ways. For example using an improved version of the ranking, namely, the inference patterns can also be ranked. The strength of the inference pattern is approximated as a value that depends on the pattern used, and the attribute used in the pattern itself. In addition, in some cases the patterns are reinforcing each-other when they are found through a different search anchors. Ranking can also be improved by using external resources like Google<sup>31</sup>, such an approximation scheme has been successfully applied in an experimental setup in matching music metadata [Gligorov et al., 2007]. This will be discussed extensively in Chapter 7.

The use of structured background knowledge has the potential to improve the intelligence of the current search engines, so that they appear as searching semantically. When queried with *Guitar* the semantic matching pointed to the *Rock* class at the providers, and the query *Saxophone* directed to *Blues*, etc. The classic music content provider's portals usually offer search limited to song, album or artist name, and are therefore unable to provide meaningful answer to these queries. A major benefit of this approach is that it increases the vocabulary offered to the user in expressing the search query. This benefit is two-folded, the background knowledge provides alternatives for the terminology used in the target, but also makes it possible to use words which are otherwise semantically related to the searched content.

## 6.4 Conclusions

We summarize the findings in our study in the following conclusions:

- Combining relations in the background knowledge has clear benefit for the matching. The experiments in the first case study revealed that combining relations of the same type provided the first major jump in the matching performance (which was Experiment 3 where the transitive chaining was first applied), and even better performance was obtained when combining relations of different types in a sensible manner, which were the Experiments 4 and 5.
- Domain-specific relations in the background knowledge can be successfully applied in different realistic matching tasks in various domains. The two case studies showed this on examples from two distant and disparate domains - medical and music, while solving realistic matching tasks - document reclassification and music search on the Internet.

---

<sup>31</sup><http://www.google.com/>

Finally, both case studies reinforced the findings of the previous chapters that using the background knowledge provides means to find matches which are otherwise missed by the traditional direct matching techniques. In the first case study we illustrated the example match **Temporal lobe**  $\xrightarrow{\cong}$  **Head**, and in the second case study the match **Guitar**  $\longrightarrow$  **Rock**



# Chapter 7

## Using unstructured background knowledge

The ontology matching problem becomes even harder in domains where concepts are inherently vague and ill-defined, and cannot be given a crisp definition. A notion of approximate concept matching is required in such domains, but until now, no such notion is available.

The first contribution of this chapter is a definition for *approximate matching between concepts*. Roughly, a matching between two concepts is decomposed into a number of submatches, and a *sloppiness value* determines the fraction of these submatches that can be ignored when establishing the matching.

A potential problem of such a definition is that with an increasing sloppiness value, it will gradually allow matches between any two arbitrary concepts. To improve on this trivial behaviour, we need to design a heuristic weighting which minimises the sloppiness required to conclude desirable matches, and at the same time maximises the sloppiness required to conclude undesirable matches. The second contribution of this chapter is to show that unstructured information as found on the Web can be used as a similarity measure that has exactly these desirable properties. Effectively, we will show that the Web (or least: the Web as indexed by state of the art search engines) can be used as background knowledge in ontology matching.

We show in this chapter, first, that background knowledge can be applied not only to discover ontology matches but also to improve an approximate matching discovery, and second, the background knowledge is not necessarily and ontology - in this case, it is the content of the web, as indexed by a search engine. The approach we present here makes use of the huge amount of

knowledge that is implicit in the current Web, and exploits this knowledge as a heuristic for establishing approximate matches between ill-defined concepts. This chapter is based on the work published in [Aleksovski et al., 2004] and [Gligorov et al., 2007].

## 7.1 Introduction & motivation

### 7.1.1 Introduction

In many realistic domains it is impossible to give precise concept definitions, and consequently no crisp notion of concept equivalence exists. Below we will illustrate this in the music-domain, where musical genres are inherently imprecise. Such imprecision is a fundamental aspect of many other domains as well. Ontology matching must then be redefined to finding a concept with the closest meaning in the other schema when an equivalent one does not exist. We then require mechanisms that are able to find approximate correspondences rather than exact ones.

The first contribution of this chapter is to define a notion of *approximate ontology matching* between inherently imprecise domain concepts (section 7.2). In section 7.3 we refine this definition with a weighting function to ensure that the approximation method does not simply allow any matches, but that correct approximations are favoured over incorrect ones. As the second main contribution of this chapter, in section 7.3.3, we instantiate this weighting function with a Google-based scheme, and show in section 7.4 through experiments in the music domain that this weighting scheme has indeed the desired behaviour of increasing recall without losing precision (i.e. favouring correct approximations over incorrect ones).

Before moving to the technical part of the chapter, we first briefly discuss the domain of musical genres, and will argue why this is an appropriate domain for investigating techniques for approximate ontology matching.

### 7.1.2 Internet Music Schemas

As we already discussed in 6.3, music genres and styles are intrinsically ill-defined, see [Aucouturier and Pachet, 2003, Pachet and Cazaly, 2000]. There is no single authority that can decide for a music entity which genre or style it belongs to. For example, it's becoming increasingly difficult to categorise the newly emerging musical styles that incorporate features from multiple

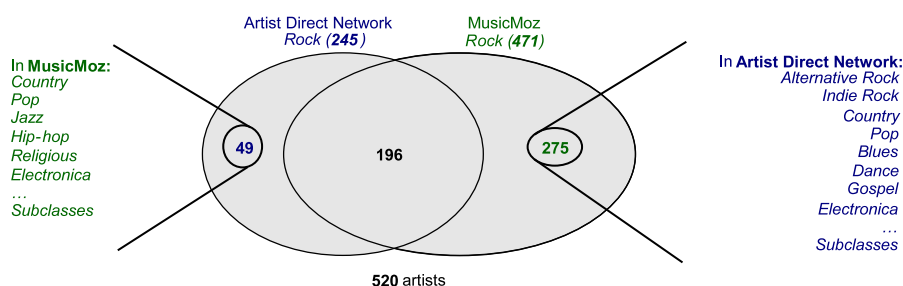


Figure 7.1: Low semantic agreement between ADN and MM.

genres. Also, the attempts to classify particular musicians in a single genre are sometimes ill-founded as they may produce music in a variety of genres over time or even within a single piece.

There are no objective criteria that sharply define music classes. Genre is not precisely defined. When asked, people will classify the same music entity in different genres, with an agreement of only in the 30-40% region. As a result, different providers often classify the same music entities (artists, albums, songs...) differently. Widely used terms like Pop and Rock do not denote the same sets of artists at different portals, [Aucouturier and Pachat, 2003]. That is also the case for even more specific styles of music like Speed Metal.

In our experiments when testing with instance data, we restricted to the artists shared by MusicMoz and Artist Direct Network, i.e. artists that are present and classified in both portals. In the sequel we refer to them as MM and ADN, respectively. As an example, from the class named **Rock** (including its subclasses) in MM there are 471 shared classified artists, in ADN there are 245, and 196 shared artists are classified under **Rock** in both of them. Hence, from all the artists classified under **Rock** in at least one of the two portals, only about 38% (196 out of 520) is classified under **Rock** in both portals, see Figure 7.1.

This example shows that there is a high degree of imprecision in the music domain. Therefore we expect that reasoning methods that look for exact matches are not useful, and approximate methods are more appropriate.

**Representation of hierarchical Music Schema** As with any semantic concept, musical genres can in principle be defined either intensionally (as a set of rules that define when an entity belongs to the genre or not), or extensionally (as the set of all music entities that belong to the genre). With

rare exceptions such as Music Genome<sup>1</sup> and Wikipedia<sup>2</sup>, who aim to provide intensional definitions of musical genres, the extensional approach is the most widely used in practice. Hence, for our purpose, we assume an extensional treatment for the genres and styles as sets of music entities. Consequently, a music classification is a collection of music concepts described with English language terms, where instances are being classified. It can be modelled as a concept hierarchy.

However, it is not sufficient to use only the concept labels to identify the concepts, since, their position in the schemas influences their meaning as well. Figure 7.2 illustrates this with an example from existing music schemas. Although the labels are equivalent (namely **Experimental**), they represent different classes. We adopt the approach proposed in [Bouquet et al., 2003] to make the meanings explicit by conjoining the concepts with their superconcepts in the hierarchy. This then makes the meaning of the two concepts in the example explicit as

Electronic  $\cap$  Experimental  
Jazz  $\cap$  Big Band  $\cap$  Experimental

**Data Selection** As experimental data we used the seven music metadata schemas extracted from different Internet music providers which we described in detail in the previous chapter in Section 6.3.2. Briefly, the metadata schemas are not offered explicitly, but we extracted them from the implicit structure of the navigation paths together with the labelling on the links on the web pages. Also, we have extracted the music schema present in the free online encyclopedia Wikipedia.

## 7.2 An approximation method for ontology matching

We phrase our problem as finding subsumption relations  $A \subseteq B$  between concepts from separate hierarchies (with finding equivalences as an obvious special case of mutual subsumptions). The representation of concepts as a conjunction of propositional symbols described in the previous section implies

---

<sup>1</sup><http://www.pandora.com/mgp.shtml>

<sup>2</sup>[http://en.wikipedia.org/wiki/Music\\_genre](http://en.wikipedia.org/wiki/Music_genre)

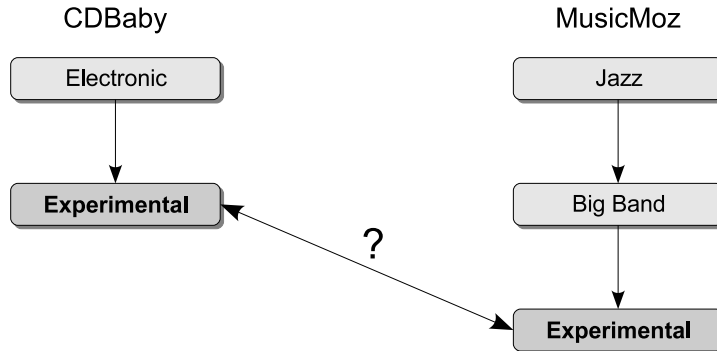


Figure 7.2: Two music genres: Although the labels are equivalent, **Experimental**, they represent different classes.

that concepts have the form  $B = B_1 \cap \dots \cap B_k$ <sup>3</sup>. This means that the subsumption check

$$A \subseteq B \quad (7.1)$$

can be split into a set of *subproblems*:

$$\bigwedge_i (A \subseteq B_i) \quad (7.2)$$

with each subproblem checking if  $A$  is a subclass of one of the conjuncts  $B_i$ . The original subsumption problem (7.1) is satisfied if and only if all the subproblems from (7.2) are satisfied.

This approach is independent from the choice of solver to use for solving the individual subproblems  $A \subseteq B_i$ . This might be an intensional reasoner (using axioms for  $A$  and  $B_i$ ), or an extensional checker (checking the instance sets of  $A$  and  $B_i$ ), or a simple lexical approach (using the textual descriptive labels of  $A$  and  $B_i$ ). In our experiments later in this chapter, we will in fact use the lexical approach, but our approach to approximate ontology matching is independent of this choice.

Now we introduce the idea of approximation. In our approximation, we allow a few of the subproblems from (7.2) to be unsatisfiable, while still declaring the original problem (7.1) satisfiable. The (relative) number of satisfiable subproblems is a measure of how strongly the subclass relation between the two given formulas hold. If for only a few of the subproblems the relation (7.2) doesn't hold, we may say that the relation (7.1) *almost*

<sup>3</sup>Remember that we use the interpretation of the concepts as sets and consequently we replace conjunction by intersection relation, and implication by a subset relation.



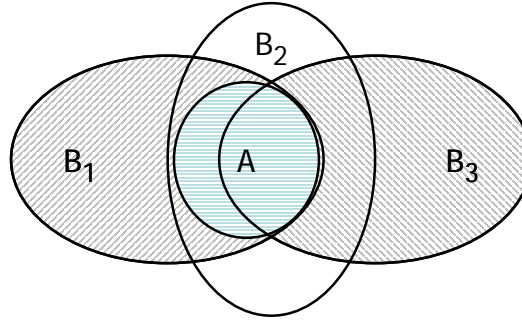


Figure 7.3: Example of an approximate subsumption relationship:  $A \subseteq B$  with  $B = B_1 \cap B_2 \cap B_3$ .

*holds.* We can express the strength at which the relation (1) holds as the ratio between the number of false subproblems (subproblems for which the subclass relation doesn't hold) and the total number of subproblems. We call this ratio the *sloppiness* and we use the letter  $s$  to denote its value:

$$s(A \subseteq B) = \frac{|\{i : A \not\subseteq B_i\}|_4}{k} \quad (7.3)$$

Here  $|\{i : A \not\subseteq B_i\}|$  denotes the number of unsatisfied subproblems that are ignored, and  $k$  is the total number of subproblems. Since all subproblems are treated equally, we refer to this sloppiness value as the *uniform weighting*.

In the example of Figure 7.3, the subsumption  $A \subseteq B_1 \cap B_2 \cap B_3$  is not classically valid, because  $A \not\subseteq B_3$ , but it is valid at a sloppiness level of  $1/3$  (or higher).

Some properties that can be easily seen are:

**Property 1.** *Only classically valid subsumptions hold at  $s = 0$ .*

**Property 2.** *Any subsumption holds at  $s = 1$ .*

From these two properties it follows that our approximation only makes sense for conjunctive formulae  $B_1 \dots B_k$  with  $k > 1$ . A trivial conjunctive formula with only one conjunct is not approximable: the only sloppiness values for such a formula are either 0 or 1, limiting us to either classical subsumption or trivial acceptance of the subsumption. But such a formula would correspond to a hierarchy of depth 1, which is a case not very interesting for our problem anyway.

---

<sup>4</sup>We will often omit the argument to  $s$  if this is clear from the context, and simply speak of the sloppiness value  $s$ .

**Property 3.** *The set of subsumptions that hold at a given sloppiness value grows monotonically with increasing sloppiness value.*

Properties 1-3 together tell us that at small  $s$ -values, the approximation will be too strict, and no (actually: only classically valid) matches will be found; while at large  $s$ -values, the approximation will be too loose, and many invalid matches (actually: all combinations) will be found.

In the next subsection, we will discuss how we can influence the approximation level (the  $s$ -values) at which desirable and undesirable matches are found.

## 7.3 Heuristic weighting

### 7.3.1 Defining weights

We can influence the sloppiness level at which a match is discovered with a heuristic *weighting function*: each subproblem  $A \subseteq B_i$  is given a weight  $w_i$ , and the *heuristic sloppiness* value  $\tilde{s}(A \subseteq B)$  is then the summed weight of all the subproblems that needed to be ignored for the subsumption to be accepted:

$$\tilde{s}(A \subseteq B) = \sum_{\{i:A \not\subseteq B_i\}} w_i.$$

Notice that the non-heuristic  $s$ -value (the uniform weighting) is a special case of the heuristic  $\tilde{s}$ -value, namely with equal values for all  $w_i$ .

### 7.3.2 Choosing Weights

How should we make a heuristic weighting function, and how should we judge if one weighting heuristic is better than another? As before, let  $A \subseteq B$  with  $B = B_1 \cap B_2 \cap B_3$  be a potential ontology match (= cross-hierarchy subsumption relation) that does not hold classically. To judge the quality of a weighting heuristic, we must distinguish two cases:

**desirable matches:** although  $A \subseteq B$  does not hold classically, it might nevertheless be a useful matching. (This might be known from intended (informal) meaning of  $A$  and  $B$ , or because we have a gold standard that contains this match). In this case, we would want  $A \subseteq B$  to be derived at a *low* level of sloppiness (i.e. by ignoring only a small number of well-chosen subproblems)

**undesirable matches:** conversely, we might know from the intended meaning of  $A$  and  $B$  that indeed  $A \subseteq B$  should not be derived. In this case, we would want  $A \subseteq B$  to be derived only at a very high level of sloppiness (i.e. only after ignoring many subproblems).

Put in other words, a heuristic weighting function should minimise the sloppiness required to derive desirable matches, and maximise the sloppiness required to derive undesirable matches. This would ensure that when gradually increasing the allowed sloppiness level, we begin to discover desirable matches quickly, while only including undesirable matches very late in the process. This would have the effect that *when increasing the  $s$ -value, we have an early increase of recall, but a late decrease of precision.*

It is easily seen that the uniform weighting scheme from formula (7.3) amounts to choosing equal values for all  $w_i$  (namely  $1/k$ ), and hence to making a random choice for subproblems to ignore. In general, however, the subterms  $B_i$  have different significance in capturing the meaning of  $B$ . We would expect any reasonable heuristic to improve over a random choice, and ignoring more significant subterms should go with a penalty, i.e. should require a higher  $w_i$  for such significant subterms. A requirement on an informed heuristic form choosing the weights  $w_i$  is therefore:

**Requirement 1.** *The heuristic weights  $w_i$  should be chosen in such a way that:*

- *if  $A \subseteq B$  is a desirable match,  
then  $\tilde{s}(A \subseteq B) < s(A \subseteq B)$*
- *if  $A \subseteq B$  is an undesirable match,  
then  $\tilde{s}(A \subseteq B) > s(A \subseteq B)$ .*

Consider again the example of Figure 7.3, and let  $A \subseteq B$  be a desirable match. Suppose that we heuristically choose the following weights:  $w_1 = 0.7, w_2 = 0.2, w_3 = 0.1$ . Then this heuristic accepts  $A \subseteq B$  already at sloppiness level 0.1, i.e.  $\tilde{s}(A \subseteq B) = 0.1$ , since only  $A \subseteq B_3$  with weight 0.1 needs to be non-classically assumed to hold (even though classically it doesn't).

The uniform weighting scheme would result in  $s(A \subseteq B) = 1/3$ . If, as assumed,  $A \subseteq B$  is a desirable match, then the uniform weighting performs less well than the example heuristic values, since the required increase in recall (i.e. accepting  $A \subseteq B$ ) is only achieved at a higher sloppiness value

(1/3 instead of 0.1). If on the other hand  $A \subseteq B$  would be an undesirable match, the uniform weighting would be preferred.

Clearly, the recall and precision of the the above approximation matching approach relies on choosing the right weights  $w_i$ .

The main intuition behind the choice for a good weighting function is that the weight assigned to the subproblem  $A \subseteq B_i$  should reflect how much information the concept  $B_i$  provides about the concept  $B$ . The level of *informativeness* can be observed as a “semantic closeness” between the concepts  $B_i$  and  $B$ . Intuitively, a concept  $B_i$  that is semantically close to  $B$  should be more relevant, and have a higher weight, than a concept that is semantically more distant. In the next section, we will consider a weighting heuristic based on such a notion of semantic distance.

### 7.3.3 Normalised Google Distance

The weighting scheme we consider in this section takes advantage of the vast knowledge available on the web by using a Google-based dissimilarity measure.

We utilise a dissimilarity measure, called Normalised Google Distance (NGD), introduced in [Cilibrasi and Vitanyi, 2007]. NGD takes advantage of the number of hits returned by Google to compute the semantic distance between concepts. The concepts are represented with their labels which are fed to the Google search engine as search terms. Given two search terms  $x$  and  $y$ , the *the normalised Google distance* between  $x$  and  $y$ ,  $\text{NGD}(x, y)$ , can be obtained as follows

$$\text{NGD}(x, y) = \frac{\max\{\log f(x), \log f(y)\} - \log f(x, y)}{\log M - \min\{\log f(x), \log f(y)\}} \quad (7.4)$$

where

$f(x)$  is the number of Google hits for the search term  $x$ ,

$f(y)$  is the number of Google hits for the search term  $y$ ,

$f(x, y)$  is the number of Google hits for the tuple of search terms  $x y$ , and

$M$  is the number of web pages indexed by Google<sup>5</sup>.

---

<sup>5</sup>Currently, the Google search engine indexes approximately ten billion pages ( $M \approx 10^{10}$ ).

Intuitively,  $\text{NGD}(x, y)$  is a measure for the symmetric conditional probability of co-occurrence of the terms  $x$  and  $y$ : given a web-page containing one of the terms  $x$  or  $y$ ,  $\text{NGD}(x, y)$  measures the probability of that web-page also containing the other term<sup>6</sup>.

**Example:** We will determine the normalised Google distance between the search terms “jazz” and “rock” that correspond to concepts  $B_{\text{jazz}}$  and  $B_{\text{rock}}$ , respectively. The number of hits for the search term “jazz” is given by  $f(\text{jazz}) = 196\,000\,000$  and for the search term “rock” by  $f(\text{rock}) = 723\,000\,000$ . Furthermore, Google returns  $f(\text{jazz}, \text{rock}) = 119\,000\,000$  web pages in which both “jazz” and “rock” occur. Therefore,  $\text{NGD}(\text{jazz}, \text{rock}) = 0.458846$ .<sup>7</sup>

### 7.3.4 Google-based weighting

For a subsumption check  $A \subseteq B$  with  $B = B_1 \cap \dots \cap B_k$ , the Google-based weighting scheme is defined as follows:

First, we compute the *normalised Google distances*.

$$d_i = \text{NGD}(B_i, B)$$

We normalise these values to the  $[0, 1]$  interval

$$d'_i = \frac{d_i}{\sum_{j=1}^k d_j} \quad (7.5)$$

Subsequently, the normalised distance values are converted into similarities

$$s_i = 1 - d'_i$$

Finally, from the similarity values the weights for the subproblems are derived.

$$w_i = \frac{s_i}{\sum_{j=1}^k s_j}$$

---

<sup>6</sup>The NGD measure assumes monotonicity of Google. In reality Google is known to show non-monotonic behaviour, i.e. adding more words in the search query may increase the number of hits instead of decrease it. Yet, such cases are exceptions and did not affect the results of our experiments

<sup>7</sup>The values for these queries were obtained by conjoining the search terms (“jazz” and “rock”) with the general scope-term “music”, in order to avoid homonym problems such as “rock” in its geological sense.

### 7.3.5 Modified Google Distance

The general NGD formula (7.4) (taken from [Cilibrasi and Vitanyi, 2007]) can be slightly simplified because of the special form of our queries. As said, we compute the NGD values using (7.4)

$$\begin{aligned} d_i &= \text{NGD}(B_i, B) \\ &= \frac{\max\{\log f(B_i), \log f(B)\} - \log f(B_i, B)}{\log M - \min\{\log f(B_i), \log f(B)\}} \end{aligned} \quad (7.6)$$

where  $f(B_i)$ ,  $f(B)$  and  $f(B_i, B)$  give the number of hits for  $B_i$ ,  $B$  and  $(B_i, B)$ , respectively.

The Google query for  $B_i$  is comprised of search terms that are also present in the Google query for  $B$ . Therefore,  $f(B_i) \geq f(B)$ . This inequality implies that

$$\begin{aligned} \max\{\log f(B_i), \log f(B)\} &= \log f(B_i) \text{ and} \\ \min\{\log f(B_i), \log f(B)\} &= \log f(B). \end{aligned}$$

For the same reason (namely that all terms from  $B_i$  also occur in  $B$ ), the Google query for  $B$  coincides with the Google query for the tuple  $(B_i, B)$  which means  $f(B_i, B) = f(B)$ .

If we consider the last few deductions we can rewrite (7.6) in the following manner

$$d_i = \text{NGD}(B_i, B) = \frac{\log f(B_i) - \log f(B)}{\log M - \log f(B)}$$

or

$$d_i = \text{NGD}(B_i, B) = \frac{N_i}{N} \quad (7.7)$$

where  $N = \log M - \log f(B)$  and  $N_i = \log f(B_i) - \log f(B)$

If we consider (7.7) we can rewrite the expressions given with (7.5) as follows:

$$d'_i = \frac{\frac{N_i}{N}}{\sum_{j=1}^k \frac{N_j}{N}} = \frac{N_i}{\sum_{j=1}^k N_j}$$

As we can see the value of  $d'_i$  does not depend on  $N$ . Therefore, we can use the following expression to compute the Normalised Google Distance

$$d_i = m\text{NGD}(B_i, B) = \log f(B_i) - \log f(B) = \log \frac{f(B_i)}{f(B)}$$

where  $m\text{NGD}$  stands for modified Normalised Google Distance. The advantage of  $m\text{NGD}$  over the NGD is that  $m\text{NGD}$  no longer depends on  $M$ , the size of the Google index, for which we would have to guess a value.

### 7.3.6 Examples

In this section we illustrate the process of approximate ontology matching with weighting. In our examples we consider the subsumption relations between two pairs of styles from MusicMoz and ArtistDirectNetwork portals, as shown in Figures 7.4 and 7.5.

**Example 1: Country and Bluegrass gospel** The first step is to transform the concepts into formulas. The individual concepts are represented with their labels, and conjoined with the representation of their parents, as discussed in section 7.1.2. This yields the following formulas representing the meaning of Country from ADN and Bluegrass Gospel from MM:

$$\begin{aligned} \text{ADN: Country} &= \text{Country} \\ \text{MM: Bluegrass Gospel} &= \text{Country} \cap \text{Bluegrass} \cap \text{Bluegrass Gospel} \end{aligned}$$

We use these formulas to test for the subsumption relation

$$\text{ADN: Country} \subseteq \text{MM: Bluegrass Gospel}.^8 \quad (7.8)$$

which is depicted in Figure 7.4. As described in section 7.2, we have to solve the following subproblems:

$$\begin{aligned} \text{Country} &\subseteq \text{Country} \\ \text{Country} &\subseteq \text{Bluegrass} \\ \text{Country} &\subseteq \text{Bluegrass Gospel} \end{aligned}$$

In order to solve the individual subproblems, we apply the following method:

- Concepts with the same label have the same meaning
- If one label is derived from another by adding extra words, we assume that the first is a more specific concept than the second, hence the second concept subsumes the first. This is a reasonable assumption because additional words usually restrict an expression's meaning.

---

<sup>8</sup>As any music-lover will know, this subsumption is actually false, and hence constitutes an undesirable match.

Given this, we obtain the following results for the subproblems

$$\begin{aligned} \text{Country} \subseteq \text{Country} & : \text{true (Country on both sides)} \\ \text{Country} \subseteq \text{Bluegrass} & : \text{false} \\ \text{Country} \subseteq \text{Bluegrass Gospel} & : \text{false} \end{aligned}$$

Using the uniform weighting scheme, subsumption (7.8) is acceptable at a sloppiness-level of 0.66, since 2 out of 3 subproblems are found not to hold, and must be ignored in order for the subsumption to go through

Next, we apply the Google-based weighting scheme on the same set of subproblems. We compute the NGD values as described in section 7.3.4:

$$\begin{aligned} d_1 &= \text{NGD}(\text{Country}, \text{Country} \cap \text{Bluegrass} \cap \text{Bluegrass Gospel}) \\ d_2 &= \text{NGD}(\text{Bluegrass}, \text{Country} \cap \text{Bluegrass} \cap \text{Bluegrass Gospel}) \\ d_3 &= \text{NGD}(\text{Bluegrass Gospel}, \text{Country} \cap \text{Bluegrass} \cap \text{Bluegrass Gospel}) \end{aligned}$$

After providing each of the terms with the scope-term “music”, the Google queries “Country” music, “Bluegrass” music, “Bluegrass Gospel” music and “Country” “Bluegrass” “Bluegrass Gospel” music return 467 000 000, 24 000 000, 338 000 and 261 000 hits, respectively. Consequently, we find the following  $m\text{NGD}$  distances:

$$\begin{aligned} d_1 &= 7.44147 \\ d_2 &= 4.39942 \\ d_3 &= 0.164622. \end{aligned}$$

Using these values we derive the weights for the subproblems, as described in section 7.3.4:

$$\begin{aligned} w_1 &= 0.190081 \\ w_2 &= 0.336629 \\ w_3 &= 0.493144 \end{aligned}$$

Since the last two of the three subproblems must be ignored for the match to go through, the heuristic  $\tilde{s}$ -value is 0.81. Hence, in this example,  $\tilde{s}$ -value  $>$   $s$ -value, which, since the match is actually undesirable, is an improvement (of 15 %-points) over the uniform weighting, in line with the requirement stated in section 7.3.1.

**Example 2: Black metal and Heavy metal** The previous example illustrated the behaviour of the weighting on an undesired match. In this example



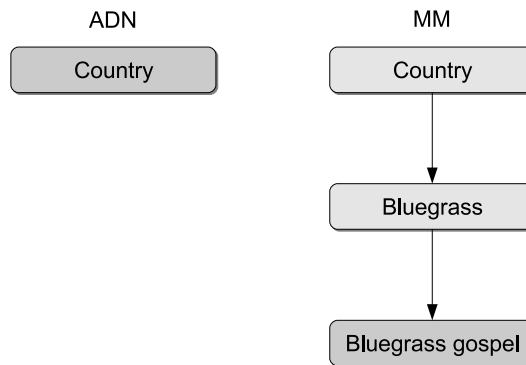


Figure 7.4: Matching concepts Country and Bluegrass gospel

we examine the weighting behaviour on a desired match. Figure 7.5 shows the following matching problem:

$$\text{ADN: Black metal} \subseteq \text{MM: Heavy metal.}^9 \quad (7.9)$$

The separate subproblems in this case are:

$$\begin{aligned} \text{Rock} \cap \text{Heavy metal} \cap \text{Black metal} &\subseteq \text{Rock} \\ \text{Rock} \cap \text{Heavy metal} \cap \text{Black metal} &\subseteq \text{Hard} \\ \text{Rock} \cap \text{Heavy metal} \cap \text{Black metal} &\subseteq \text{Black metal} \end{aligned}$$

The first and the third subsumptions are correct established lexically, but the second is found false. As a consequence the uniform weighting scheme would need a sloppiness-level of  $s = 0.33$  to establish (7.9). Using NGD we obtain the following weight values for each of the subproblems:

$$\begin{aligned} w_1 &= 0.25462 \\ w_2 &= 0.25588 \\ w_3 &= 0.48950. \end{aligned}$$

This allows (7.9) to be established at a heuristic sloppiness-level of  $\tilde{s} = 0.25588$ . This is 8 %-points lower than the required  $s$ -value, and hence an improvement, since (7.9) is a desired match.

### 7.3.7 Related work

Our Google-based weighting method resembles recent work in the field of *Knowledge Acquisition* which exploits the Web as a source for discovering

<sup>9</sup>Black metal is widely accepted as being a substyle of Heavy metal

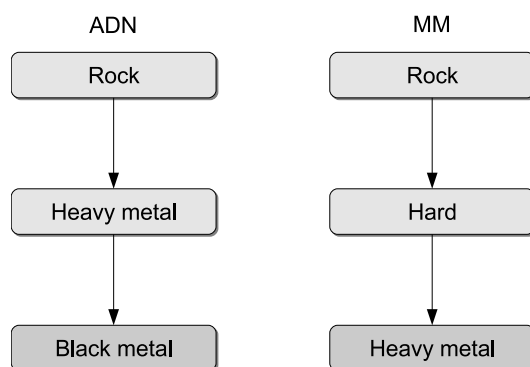


Figure 7.5: Matching concepts **Black metal** and **Heavy metal**

new knowledge, for example to discover relations between concepts in various ways: using the number of hits when querying a search engine like Google, by analysing co-occurrence of concepts within large text corpora, or by exploiting patterns to construct queries to check for a relation (see [Hearst, 1992, Agichtein, 2005, Crescenzi and Mecca, 2004, Cilibrasi and Vitanyi, 2007] for recent work).

Other attempts exist to deal with vaguely defined domain concepts, most notably by using fuzzy-logic representations (e.g. [Straccia, 2006]), and more recently by using rough sets [Schlobach et al., 2007]. An inherent price of these approaches is that they require significant extra modelling effort to capture the imprecision of the concepts, either in terms of fuzzy membership functions or in terms of rough-set upper- and lower-bounds. The advantage of our approximation approach is that it applies directly to semantically lightweight hierarchies typically found in imprecise domains without any further need for complicated and expensive domain modelling.

We also point out that the “confidence-factor” which is part of the format used by the Ontology Alignment Evaluation Initiative<sup>10</sup> only expresses the degree of confidence in the truth of a *crisp* matching relation, and does not represent an approximation of such a matching relation.

## 7.4 Experiments

To validate our approach of improving approximation by weighting, we conducted experiments using data from the music domain. In this section, we describe the experiments and summarise and discuss the results.

<sup>10</sup><http://oaei.ontologymatching.org/>

### 7.4.1 Experimental setup

**Goal of the experiment:** In our experiment, we want to measure both if approximation improves over simple classical matching (first goal), and if approximation with heuristic weighting improves over uniform weighting (second goal).

**Data acquisition:** We used genre hierarchies that we extracted from the meta-data schemas underlying the classifications of the Artist Direct Network (ADN), MusicMoz (MM) and Wikipedia music portals, as discussed in section 7.1.2.<sup>11</sup> One of the concept was always from MM, and the other concept in 30% of the cases from ADN and 70% of the cases from Wikipedia

**Construction of a Gold Standard:** We randomly selected 50 pairs of concepts, and manually assessed whether a match (= a cross-hierarchy subsumption relation) between the concepts within each selected pair holds or not. In other words, we classified each of the 50 pairs as either a *desirable* or an *undesirable* matching relation, as defined in section 7.3.2. This yielded 9 desirable matches and 41 undesirable matches.<sup>12</sup>

**Evaluation criteria:** For measuring the first goal (improvement of approximate matching over classical matching), we use standard recall and precision measures: heuristic matching is better than classical matching if recall improves (more desirable matches are found), while precision does not decrease much (not many undesirable matches are found). For measuring the second goal (improvement of heuristic weighting over uniform weighting), we use Requirement 1 from section 7.3.2. Heuristic weighting is better than uniform weighting if desirable matches are found with a lower heuristic  $\tilde{s}$ -value than the uniform  $s$ -value, and conversely for undesirable matches.

**Choice of subproblem solver:** Remember that we regard the ontology matches as cross-hierarchy subsumptions, and that our approximation method reduces subsumption-queries  $A \subseteq B$  to a set of subproblems  $A \subseteq B_i$ . Of course, each of these atomic subproblems still needs to be solved. As in section 7.3.6, we apply a simple lexical method that establishes whether  $A$  is (strictly) subsumed by  $B_i$  by checking whether the set of words in label of  $A$  is (properly) contained in the set of words of the label of  $B_i$  (since additional words usually restrict an expression's meaning).

---

<sup>11</sup>We selected these particular genre hierarchies because they contained enough additional information for us to construct a Gold Standard.

<sup>12</sup>It is probabilistically plausible that of set of randomly drawn cross-hierarchy pairs, the large majority are undesirable matches.

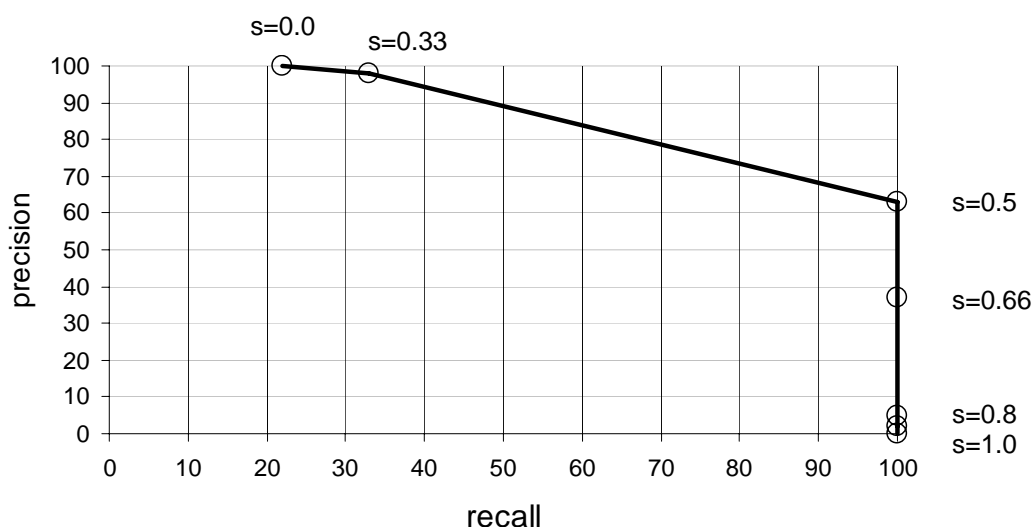


Figure 7.6: **Recall/Precision** graph of approximate matching with uniform weighting at increasing sloppiness-levels. Exact matching happens at  $s=0$ .

## 7.4.2 Comparing approximate and exact matching

When comparing approximate matching with exact matching, we cannot simply compare two scores. After all, the behaviour of approximate matching is a function of the sloppiness-level  $s$ .

Figure 7.6 shows a recall-precision plot of uniformly weighted approximate matching. Since exact matching corresponds to approximate matching at  $s = 0$ , the graph also shows the precision and recall of exact matching: 22% recall at 100% precision. This score is to be expected of the simple lexical matcher that we employ to solve subproblems: lexical matching will fail to find many semantically desirable matches (hence the low recall of 22%), but when it finds a match, it is indeed a correct one (precision of 100%).

As expected of approximate matching, increasing values of  $s$  increase recall and decrease precision. (Figure 7.6 quantitatively shows what was already qualitatively predicted by properties 1 and 2 from section 7.2).

Concerning the comparison with exact matching, Figure 7.6 shows that a slight increase in  $s$ -levels is indeed useful (a jump in recall from 22% to 33% at a negligible loss of precision when increase  $s$  from 0.2 to 0.33). The graph also shows that there is no point in increasing the uniform  $s$ -value beyond 0.5, since at that point full recall is achieved. However, this has happened at the cost of a significant loss in precision (down to 63% at  $s = 0.5$ ).

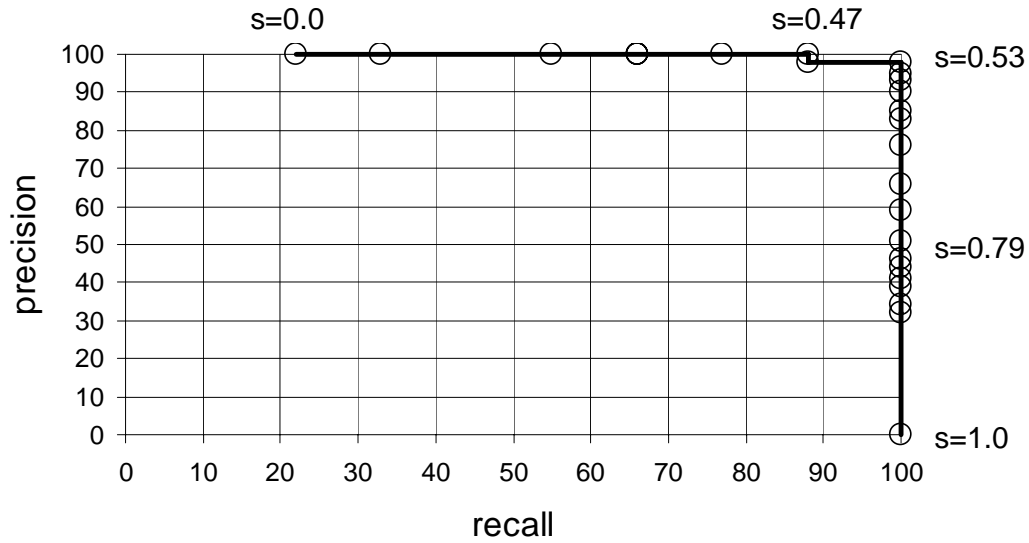


Figure 7.7: **Recall/Precision** graph of approximate matching with Google weighting at increasing sloppiness-levels.

Overall, Figure 7.6 tells us that approximate matching at small  $s$ -values does indeed usefully improve over exact matching, even when randomly choosing the subproblems to be approximated (= the uniform weighting).

In the next section, we will evaluate to what extent an informed weighting heuristic can improve over this.

### 7.4.3 Google-based weighting experiments

The behaviour of the Google-weighted approximate matching is shown in the recall-precision plot of Figure 7.7. This is near perfect recall-precision plot, which shows that we can increase the  $\tilde{s}$ -values (and hence increase the recall) until we have achieved perfect recall, all at the cost of only a negligible loss in precision of 2%, at  $\tilde{s} = 0.53$ . The precision only starts to drop significantly after a perfect recall has been achieved. This plot compares very favourably with that of the uniform weighting scheme in Figure 7.6, where a perfect recall could only be achieved at a disappointing 63% precision.

Interestingly enough, both the uniform and the Google weighting achieve 100% recall around the  $\tilde{s} = s = 0.5$  level (i.e. ignoring about half the subproblems), but apparently the Google weighting is very successful at maintaining exactly those subproblems which prevent a drop in precision (while the uniform weighting simply ignores random subproblems).

	<b>Total</b>	<b>Better</b>	<b>Equal</b>	<b>Worse</b>
Undesirable matches	41	39	1	1
Desirable matches	9	5	3	1
Total	50	44(88%)	4(8%)	2(4%)

Figure 7.8: NGD weighting compared to uniform weighting. “Better” and “Worse” mean NGD weighting is better (worse) than uniform weighting.

One observation which questions the stability of our results is that after the  $\tilde{s} = 0.5$ -level, the precision starts to drop very sharply (to 83% at  $\tilde{s} = 0.75$  and even to 51% at  $\tilde{s} = 0.79$ ). This suggests that on other data-sets, the near perfect score at  $\tilde{s} = 0.53$  may not be repeated. Further experiments on other data-sets are required to verify this.

An overall summary of the results of our experiments is given in Figure 7.8. Informed weighting yielded an improvement over uninformed weighting in almost all cases. The main gain is in a better performance on undesirable matches (i.e. improving precision).

The most detailed analysis of our experimental data is shown in Figures 7.9 and 7.10. There we plot for all 41 undesirable matches (Figure 7.9) and for all 9 desirable matches (Figure 7.10) the minimal  $s$ - and  $\tilde{s}$ -values at which these matches are found. Figure 7.9 shows that for almost all undesirable matches, the informed  $\tilde{s}$ -values are higher than the  $s$ -value, meaning that the informed weighting scheme is more resistant to accepting such undesirable matches (explaining its dramatically better precision scores in the preceding recall/precision plots). Figure 7.10 shows that for the desirable matches, both weighting schemes behave roughly equally well.

Taken together, we can conclude that the main gain of the informed weighting scheme is that it manages to avoid a drop in precision while maintaining the high recall-levels of the uninformed scheme.

Figure 7.9 also shows that the lion-share of the gains by the informed weighting are made at the  $s = 0.5$  level, where the uninformed weighting accepts an incorrect match. These are often cases with labels of length 2 (such as “hard rock”  $\subseteq$  “gospel rock”). In this example, “gospel” is the most informative term, but since the uninformed weighting randomly chooses a subproblem to ignore, it may choose to ignore “gospel”, ending up with “hard rock”  $\subseteq$  “rock” as the only subproblem, which does indeed hold. The informed Google weighting would instead realise that “gospel” is the most informative term, hence choose to ignore “rock” instead, ending up with “hard rock”  $\subseteq$  “gospel” as the remaining subproblem, and correctly refusing

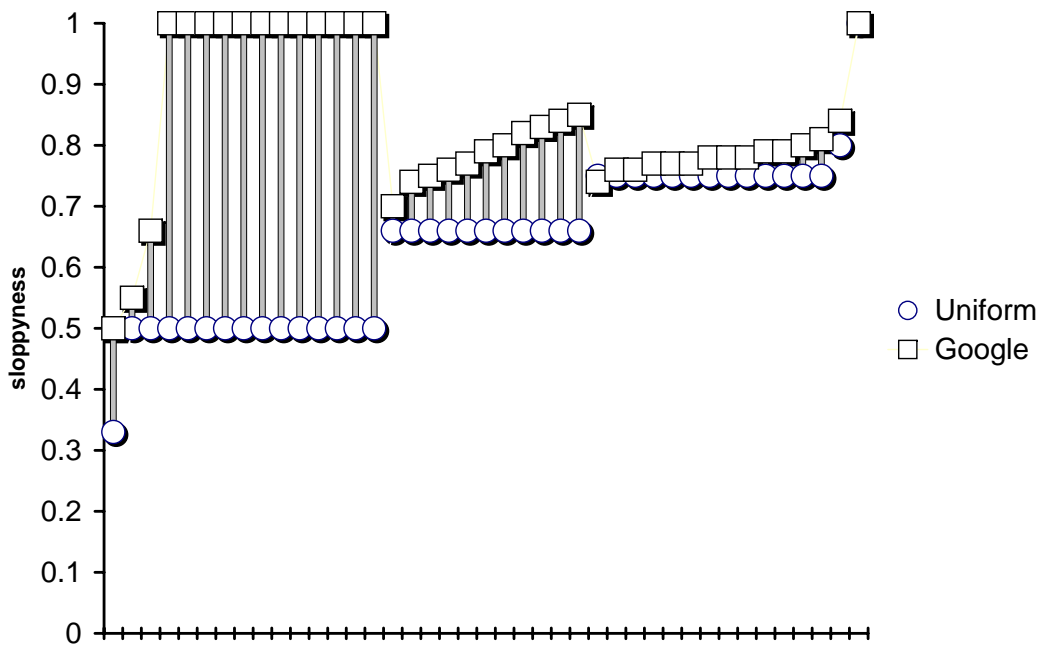


Figure 7.9: Google-based versus uniform weighting on undesirable matches

to accept this match. Note that this is not simply a matter of preferring adjectives over nouns, consider for example the case “Country Gospel”  $\subseteq$  “Christian Gospel”, which is an intended matching, and can only be established by preferring the noun “Gospel” over the adjective “Christian”. A truly semantically informed weighting scheme is required, no simple lexical fix will do.

## 7.5 Conclusion

In this chapter, we have addressed the problem of discovering approximate matching relations between concepts from different concept hierarchies. Such approximate matching relations are required in many domains where concepts are ill-defined, and any attempt to find precise equivalences (or even precise subsumptions) will fail because of the imprecise nature of the concepts. Our method is directly applicable to the lightweight hierarchies found in practice in imprecise domains.

We have given a declarative definition for approximate ontology-matching with a variable degree of approximation. We have shown how this approximation degree (for which we used the term sloppiness level) can be influenced

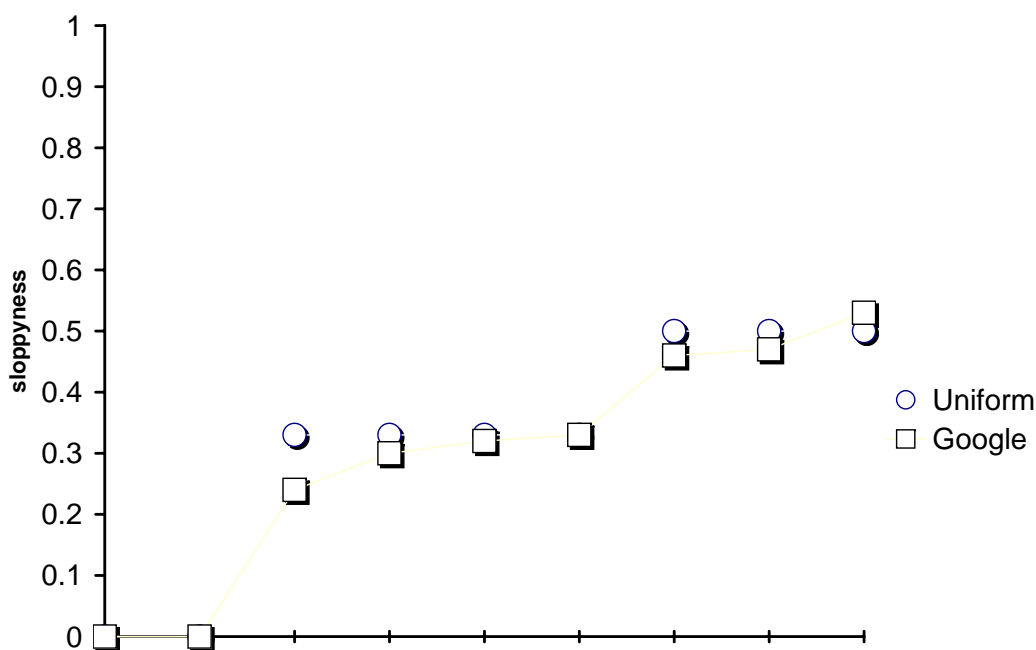


Figure 7.10: Google-based versus uniform weighting on desirable matches

by a semantic similarity measure that we derive from the Normalised Google Distance.

In order to validate our theoretical proposal, we harvested from the Web realistic concept hierarchies that represented musical genres, we constructed a small Gold Standard corpus of matching candidates, and we performed experiments to test the precision and recall of our approximation method. Our results show that the Google-based semantic measure significantly outperforms an uninformed measure.

This approach is entirely independent from the algorithm used to establish the submatches that together build up an approximate matching. In the music domain, we have used simple lexical techniques to establish these submatches, but this can be replaced with more complicated matching techniques, while the essential idea of our sloppiness value and the weighting function can still be applied unchanged.

In very general terms, this approximation method makes use of the huge amount of background knowledge that is implicit in the current Web, and exploits this knowledge as a heuristic for establishing approximate matches between ill-defined concepts.





# Chapter 8

## Conclusions

In this chapter we will summarize the findings of this thesis in terms of answering our main research question: *How can Background Knowledge be used in ontology matching?*. We conclude that using background knowledge is practically feasible, and it provides a clearly useful additional approach to the set of ontology-matching techniques.

### 8.1 Summary of main findings

Our first research question was: *1. Can the use of background knowledge contribute to the state-of-the-art ontology matching tools?* Using semantically rich background knowledge can produce a better matching than the state-of-the-art matching tools on a case study which is representative of many realistic matching cases, Chapter 4. Such a matcher using background knowledge can find matches missed by other techniques, and can compete with them on recall and precision. When there is no terminological, instance or structural match between the matching ontologies, the use of background knowledge is the only way to find the matches. Given that comprehensive ontologies are increasingly becoming available in various domains, these positive results make this approach very appealing for the newly developed matching tools.

Our second research question was: *2. What is the benefit of using multiple ontologies as background knowledge simultaneously?* Multiple background knowledge ontologies can be used together for cumulative benefit in the matching. When adding more background ontologies, the recall increases monotonically while the precision changes depending on the quality of the background knowledge. Ontologies which are created by domain experts and

which can be considered high-quality result in high precision, while ontologies of an unknown origin, which can be considered low-quality, result in low precision. High-quality background ontologies in a specific domain result in very high precision matches, and in broader domains the precision drops due to contextual problems, when there are multiple possibilities to interpret the relations in the ontologies it may lead to wrong matches, Chapter 5. The general-knowledge background ontologies in Chapter 5 resulted in precision of 75%, as compared to 95% in intensive care and 100% in the anatomy domain, reported in Chapters 4 and 6 respectively. Besides the contextual problems (the different semantic interpretations), small amounts of wrong matches were induced due to errors in the ontologies. On the other hand, the low-quality background knowledge ontologies resulted in a precision below 40%. The cause of the wrong matches was faulty relations in their content which clearly distinguished them from the high-quality ontologies.

Our third research question was: *3. How does a combination of different domain relations in the background knowledge contribute to the matching performance?* Combining relations of different types from the background knowledge can considerably improve the matching results as compared to matching without this combination, Chapter 6. However, obtaining this benefit requires prior analysis to determine how to combine the relations in the background knowledge. We note that a jump in the recall can be first observed when the relations of a single type are chained transitively, and then when relations of different types are combined we obtain the highest recall increase. But then, carefully chosen combinations of relations keep the precision at 100%, while the other combinations besides the additional correct matches produce some incorrect as well, and then reduce the precision. It is not trivial to choose which option is better. As discussed in Chapter 3, it entirely depends on the application to choose the favored precision/recall combination.

Chapter 5 revealed that the properties of the ontologies used as background knowledge influence the quality of the matches that are found. Unfortunately, most influence is exerted by properties that can be until now only reliably determined by manual inspection. They concern the correctness of the knowledge modeled with the ontology, where this correctness is also relative to a given context (within the domain of the matching ontologies).

Our fourth research question was: *4. How flexible is the scheme of using background knowledge in ontology matching?* In Chapter 7 we have shown that besides structured ontologies, it is also possible to effectively use unstructured information as background knowledge in ontology matching. We

have shown how a statistical measure of semantic distance can be used as background knowledge. One way to compute such a semantic distance is by using co-occurrence of terms within a large set of documents, which will then indicate the semantic relatedness between the terms.

Besides exact matches between concepts, it is possible to define and compute approximate ontology matches, and to devise heuristics that compute good approximations. In Chapter 7 we used background knowledge in approximate matching to improve the approximation accuracy, instead of using the background knowledge to discover the matches. This shows an additional useful application of the background knowledge in the context of ontology matching: it is not only useful to discover matches, but it is also useful to help other specific tasks within the ontology matching problem.

## 8.2 Discussion on success factors

The previous section reported positive results on the use of background knowledge in finding ontology matches. We have also seen that different types of ontologies give different success rates. These results naturally lead to the question whether we can predict *which* ontologies will yield the best results in which cases. Can we formulate a proper theory that will predict this? Can we identify "success factors" that describe which properties of an ontology will determine its success or failure as background knowledge in a particular ontology-matching task? The identification of such success factors is becoming even more important when we want to automatically make the selection of ontologies to be used as background knowledge<sup>1</sup>. Unfortunately, research on using background knowledge in ontology matching is still young. As a result, neither our own work nor results found in the literature are conclusive enough to formulate a definitive theory on such success factors. However, based on the experiments performed in this thesis, we are able to tentatively identify a number of success factors. We will discuss these factors, and our current limited understanding of them, in this section, as a first step on the way to a more solid predictive theory.

**Indicators for success** In Chapter 3 we discussed how to assess the matching success. We set out there to observe the success by comparing the precision and recall (where the application of matches determines which of the two

---

<sup>1</sup>Dynamic selection of background knowledge ontologies is done in [Sabou et al., 2006] but no selection mechanism is used in the selection - any ontology with a lexical overlap is used as background knowledge

measures is favored). In addition, matching methods can be compared in two ways: first, a method can exhibit better precision and/or recall than another method, and second, a method can produce matches which combined with the matches produced by another method improve the performance of both. In the second case the two methods are not challenged against one another, but the interest is how much do they solve the problem together. When they provide complementary contributions the best performance is achieved by combining the two. We did this in Chapter 4, where we combined the application of background ontology with a simple lexical matching. It was apparent that the background knowledge found many correct matches missed by the state-of-the-art tools, but alone it did not find some matches that the simple lexical matching found.

As revealed in Chapters 4, 5 and 6, larger background ontology size is a good indication for successful application. In Chapter 5 we concluded that the quality of the background knowledge directly influences the precision, and for ontology quality no standard notion is available so far. As can be seen from these examples, in absence of strict dependencies we are inclined to observe indicators for success rather than strict conditions that guarantee the matching success.

**Size and domain coverage of background knowledge** The trivial expectation that larger size of background knowledge leads to a higher recall was confirmed by our experiments, even though it can not be characterized as a reliable success indicator. In Chapter 5 in both case studies the largest ontologies produced the highest recall - **MeSH** in the first and **Tap** in the second case study, but the large sized ontology **ICD-10** produced no matches, and the relatively small sized **Surrey** resulted in the second highest recall, making the size a weak success indicator for the recall.

Larger size means more concepts and higher chances to establish anchors and subsequently to discover matches. But the size makes sense only if there is overlap between the domain of the matching ontologies and the background knowledge domain. Trivially, medical background ontology used to match music ontologies is unlikely to provide any matches, the background knowledge must cover the matching domain, at least partially. In Chapter 6 we saw that a hierarchy of geographical places can be useful in the music domain because the names of styles and genres often include names of geographical regions and places (**American rock**, **Chicago blues**...). Again, the domain overlap is not a guarantee for success, **ICD-10** in Chapter 5 extensively described the matching domain but produced no matches. The concepts in **ICD-10** are named with longer and more detailed names, and this

lack of naming agreement prevented successful anchoring. In this case the failure was not caused simply by different naming, but the reason is in the large difference in their conceptualization<sup>2</sup>.

The ontology size is not reflected in the number of concepts alone, it is also reflected in: the number of relations - how well are the concepts interconnected, the number of relation types - how many different types of relations are used in the ontology, the number of possible useful combinations of the relations - how many additional relations are induced in total by applying the permitted combinations of relations, and also how well is the ontology populated with instance data. In Chapter 6 we showed that combining relations of different types within the background ontology is more beneficial than using relations of a single type. Such combining needs careful consideration because different combinations yield different matching performances.

Finally, the size of background knowledge is also seen in the number of background ontologies used simultaneously. Multiple background ontologies can be used cumulatively to considerably increase the recall. In Chapter 5 we analyzed the influence of using multiple background ontologies at the same time. For a specific domain multiple background ontologies result in a cumulative increase of the recall, but the ontologies also provide overlapping contribution and some matches are discovered by multiple ontologies at the same time. For more general domain, the background ontologies increase the recall monotonically with a noticeably smaller recall increase per ontology. However, the contributions of the different ontologies are nearly disjoint, i.e. they almost find no match twice. This shows an attractive behavior because it encourages the further addition of background ontologies.

**Background knowledge quality** The background knowledge quality is directly reflected in the matching precision. In Chapter 5 it appeared that low quality ontologies result in low precision, and high quality background ontologies result in high precision. Low quality ontologies produce wrong matches because of wrong relations in their content, while this is rarely the case for ontologies of high quality. They generally have very few mistakes, and the wrong matches are mainly caused by contextual differences with the matching ontologies.

**Domain generality** High-quality background ontologies in a very specific domain result in high precision, and in broader domains the precision drops

---

<sup>2</sup>This is a good example of heterogeneity in ontologies. ICD-10 relied on very different division criteria to organize its concepts in a hierarchy.

due to the possible different semantic interpretations of the encoded knowledge. The general-knowledge background ontologies in Chapter 5 resulted in precision of 75%, as compared to 95% in intensive care and 100% in the anatomy domain, reported in Chapters 4 and 6 respectively.

The ontologies in very specific domains do not have contextual issues on the terminology - a single term is not associated with multiple different meanings. Contrary, more general domains like agriculture do have contextual issues. They contain concepts with very generic meaning - **Agent**, **Product**, etc. whose semantics largely depend on the end-user's interpretation. This ambiguity is then reflected in the matching performance by reducing the precision.

**Automatic v.s. manual background knowledge selection** Automatic background knowledge selection is attractive because it requires no manual labor. Unfortunately, with respect to the precision, the ontologies are distinguished by the quality which is directly related to the interpretation of the ontology's semantics. The quality can be reliably determined only by a manual inspection which then leads us to the conclusion that manual selection is the preferred way to find high-quality background knowledge, especially when the precision is more important than the recall. If the recall is more important, then the automatic selection can still be a good choice - the experiments in Chapter 5 showed that even the low quality background ontologies can considerably increase the recall.

In addition to the success factors discussed above, we also explored other factors such as depth and degree of connectedness of the used concepts in the background hierarchy (not reported in this thesis). However, they appeared to be inconclusive about the matching success.

### 8.3 Future research directions

**Quality requirements for ontology matching.** There is an active research in the ontology matching field which produced wide variety of matching techniques and prototype matching systems. However, it remains unclear what qualities do the practical applications need from these solutions. The possibility still remains that the very simple techniques already satisfy the requirements of the practical applications, which would then question the need for better and most likely complex solutions.

**Assessment of matching performance** Assessing the performance of a matching method is becoming increasingly difficult - currently the problem is often phrased as finding any type of relatedness between the matching concepts, whereas in the past it was most commonly defined as finding pairs of equivalent matching concepts. In addition, there are other reasons for the assessment difficulty as well: experts disagree on the correctness of the matches, and the usefulness of the matches is not anticipated in the evaluation. From an application viewpoint the matches can not be observed separately, but only as a whole - the resulting set of matches. They may depend on each other and even when a match is correct it can be undesirable by the application because other discovered matches already fulfill its purpose and make it obsolete. In addition we point out that the latest trends in Information Retrieval prefer to think about the performance in terms of probability: what is the likelihood that an answer will be relevant for the asked question [Baeza-Yates et al., 1999], instead of dividing the possible results in sets of desirable and undesirable answers.

**Combine background ontologies** In our research we used multiple background ontologies by observing the union of the matches they produce. However, the background ontologies can be further combined by establishing anchors between them as well. Then a match can be discovered even in the situation when the matching concepts are anchored to two different background ontologies. This introduces various research questions - how to establish the anchors between the background ontologies, how the qualities of the anchoring among the background ontologies affects the matching performance, and does such a combination provide benefit for the matching in the first place? This approach of combining the background ontologies was used in [Sabou et al., 2006], but the effects of the combination alone still remain unknown.

**Useful parts of the background knowledge** Another interesting research topic is to identify which parts of the background knowledge are actually responsible for the matching success. There are many research questions arising here: Which properties can be used to identify the most useful background concepts? Where are they in the hierarchy of the background ontology? Do they have more relations to the other concepts than the rest of the ontology? Do they have names with more standardized use in the domain community? Do they have shorter names than the rest? This research direction will provide deeper understanding of the effects of using background knowledge. Such answers are immediately applicable - when using a large



background ontology like UMLS with millions of concepts a selection mechanism is needed to focus on the useful parts, otherwise using the entire UMLS is practically very difficult.

### **Combination of different relations in the background knowledge**

We examined how combining different relation types influence the matching performance in Chapter 6. However, this question is domain-specific and it needs further investigation for different domains. We expect that besides hierarchically organizing the concepts, in future the ontologies will provide even more value in offering a variety of useful combinations of their relation types.

# Chapter 9

## Samenvatting

De huidige informatiesystemen (IS) worden verbeterd door het gebruik van kennisbanken (KB). Simpel, KBs voegen intelligentie toe aan informatiesystemen. Het kan de kwaliteit van de zoektocht naar informatie verbeteren, helpen met de organisatie van de enorme hoeveelheid informatie, en daarbij handmatige inspanning verminderen. Omdat de kennis van een domein verschillend en divers kan zijn, mag het IS meerdere KBs op een moment gebruiken. Maar, ze zullen eerst geïntegreerd moeten zijn voordat ze gebruikt kunnen worden. Dit bijzondere integratieprobleem is de grootste uitdaging in KB-onderzoek in zijn algemeenheid geworden. Tegenwoordig zijn er veel prototype kennisbankintegratiesystemen beschikbaar die verschillende benaderingen of combinaties van benaderingen gebruiken om de problemen op te lossen.

Dit onderzoek behandelt het gebruik van achtergrondkennis ten behoeve van KB-integratie. Het bewijst dat de integratie van KBs substantieel verbeterd kan worden door het gebruik van achtergrondkennis. Deze aanpak is praktisch uitvoerbaar en zinvol. Er is duidelijk sprake van toegevoegde waarde met betrekking tot andere integratietechnieken. Het belangrijkste zijn de volgende positieve eigenschappen: de combinatie van meer achtergrondkennis resulteert in cumulatief voordeel en de combinatie van verschillende achtergrondkennistypen ook. Uiteindelijk is het succes van achtergrondkennisgebruik niet voorbehouden aan het gebruik van gestructureerde kennisbronnen. Ook ongestructureerde kennisbanken kunnen dan zeer bruikbaar zijn.



# Bibliography

- [Agichtein, 2005] Agichtein, Y. E. (2005). *Extracting relations from large text collections*. PhD thesis, Columbia University.
- [Aleksovski et al., 2006a] Aleksovski, Z., Klein, M., ten Kate, W., and van Harmelen, F. (2006a). Matching unstructured vocabularies using a background ontology. In *Proceedings of Knowledge Engineering and Knowledge Management (EKAW)*, pages 182–197.
- [Aleksovski et al., 2004] Aleksovski, Z., ten Kate, W., and van Harmelen, F. (2004). Semantic coordination: a new approximation method. In *Proceedings of Meaning Coordination and Negotiation Workshop, at ISWC 2004, Hiroshima, Japan*.
- [Aleksovski et al., 2006b] Aleksovski, Z., ten Kate, W., and van Harmelen, F. (2006b). Bridging the semantic gap - searching music on the internet by reasoning with background knowledge. In *Proceedings of Symposium on Intelligent Algorithms*.
- [Aleksovski et al., 2006c] Aleksovski, Z., ten Kate, W., and van Harmelen, F. (2006c). Exploiting the structure of background knowledge used in ontology matching. In *Ontology Matching Workshop at International Semantic Web Conference (ISWC)*.
- [Aucouturier and Pachet, 2003] Aucouturier, J.-J. and Pachet, F. (2003). Representing musical genre: A state of the art. *Journal of New Music Research 2003, Vol. 32, No. 1, pp. 83-93*.
- [Aumueller et al., 2005] Aumueller, D., Do, H.-H., Massmann, S., and Rahm, E. (2005). Schema and ontology matching with coma++. In *SIGMOD '05: Proceedings of the 2005 ACM SIGMOD international conference on Management of data*, pages 906–908, New York, NY, USA. ACM Press.

- [Baader et al., 2003] Baader, F., Calvanese, D., McGuinness, D. L., Nardi, D., and Patel-Schneider, P. F., editors (2003). *The Description Logic Handbook: Theory, Implementation, and Applications*. Cambridge University Press.
- [Baeza-Yates et al., 1999] Baeza-Yates, R., Ribeiro-Neto, B., et al. (1999). *Modern information retrieval*. Addison-Wesley Harlow, England.
- [Berners-Lee et al., 2001] Berners-Lee, T., Hendler, J., and Lassila, O. (2001). The semantic web. *Scientific American*, 284(5):28–37.
- [Bilenko et al., 2003] Bilenko, M., Mooney, R., Cohen, W., Ravikumar, P., and Fienberg, S. (2003). Adaptive name matching in information integration. *IEEE Intelligent Systems*, 18(5):16–23.
- [Bodenreider, 2004] Bodenreider, O. (2004). The unified medical language system (umls): integrating biomedical terminology. *Nucleic acids research*, pages 267–70.
- [Bodenreider et al., 1998] Bodenreider, O., Nelsok, S. J., Hole, W. T., and Chang, H. F. (1998). Beyond synonymy: Exploiting the umls semantics in mapping vocabularies. In *Proceedings AMIA Symposium*, pages 815–819.
- [Borgida and Serafini, 2003] Borgida, A. and Serafini, L. (2003). Distributed description logics: Assimilating information from peer sources. *J. Data Semantics*, 1:153–184.
- [Bosman et al., 1998] Bosman, R., Oudemans van Straaten, H., and Zandstra, D. (1998). The use of intensive care information systems alters outcome prediction. *Intensive Care Medicine*, 24(9):953–958.
- [Bouquet et al., 2003] Bouquet, P., Serafini, L., and Zanobini, S. (2003). Semantic coordination: A new approach and an application. In *International Semantic Web Conference (ISWC)*, pages 130–145.
- [Brachman, 1977] Brachman, R. J. (1977). *A Structural Paradigm for Representing Knowledge*. PhD thesis, Harvard University, USA.
- [Cilibrasi and Vitanyi, 2007] Cilibrasi, R. and Vitanyi, P. (2007). The google similarity distance. *IEEE Transactions on knowledge and data engineering*, 19(3):370–383.
- [Cimino et al., 2003] Cimino, J. J., Min, H., and Perl, Y. (2003). Consistency across the hierarchies of the umls semantic network and metathesaurus. *J. of Biomedical Informatics*, 36(6):450–461.

- [Cochran, 1977] Cochran, W. G. (1977). *Sampling Techniques*. John Wiley & Sons, Inc.
- [Codd, 1970] Codd, E. F. (1970). A relational model of data for large shared data banks. *Commun. ACM*, 13(6):377–387.
- [Cohen et al., 2003] Cohen, W. W., Ravikumar, P., and Fienberg, S. E. (2003). A comparison of string distance metrics for name-matching tasks. In *IWeb*, pages 73–78.
- [Crescenzi and Mecca, 2004] Crescenzi, V. and Mecca, G. (2004). Automatic information extraction from large websites. *J. ACM*, 51(5):731–779.
- [D.A.B. Lindberg, 1993] D.A.B. Lindberg, B.L. Humphreys, A. M. (1993). The unified medical language system. *Meth. Inform. Med.*, 32:281–291.
- [Damerau, 1964] Damerau, F. J. (1964). A technique for computer detection and correction of spelling errors. *Commun. ACM*, 7(3):171–176.
- [Davis et al., 1993] Davis, R., Shrobe, H., and Szolovits, P. (1993). What is a knowledge representation? *AI Magazine*, 14(1):17–33.
- [de Bruijn et al., 2004] de Bruijn, J., Mart??n-Recuerda, F., Manov, D., and Ehrig, M. (2004). D4.2.1 state-of-the-art-survey on ontology merging and aligning v1. SEKT Project deliverable D4.2.1.
- [de Keizer et al., 1999] de Keizer, N., Abu-Hanna, A., Cornet, R., Zwetsloot-Schonk, J., and Stoutenbeek, C. (1999). Analysis and design of an ontology for intensive care diagnoses. *Methods of Information in Medicine*, 38(2):102–112.
- [Diana and Anaiadou, 1999] Diana, M. and Anaiadou, S. (1999). Term extraction using a similarity-based approach.
- [Do et al., 2003] Do, H., Melnik, S., and Rahm, E. (2003). Comparison of schema matching evaluations. *Lecture notes in computer science*, ISSU 2593:221–237.
- [Do and Rahm, 2002] Do, H. H. and Rahm, E. (2002). Coma - a system for flexible combination of schema matching approaches. In *VLDB*, pages 610–621.
- [Ehrig and Euzenat, 2005] Ehrig, M. and Euzenat, J. (2005). Relaxed precision and recall for ontology matching. In Ashpole, B., Ehrig, M., Euzenat, J., and Stuckenschmidt, H., editors, *Proceedings of the Workshop on Integrating Ontologies*, volume 156, page 8. CEUR-WS.org.

- [Ehrig and Sure, 2005] Ehrig, M. and Sure, Y. (2005). Foam - framework for ontology alignment and mapping; results of the ontology alignment initiative. In Ashpole, B., Ehrig, M., Euzenat, J., and Stuckenschmidt, H., editors, *Proceedings of the Workshop on Integrating Ontologies*, volume 156, pages 72–76. CEUR-WS.org.
- [Euzenat, 2007] Euzenat, J. (2007). Semantic precision and recall for ontology alignment evaluation. In *Proceedings of International Joint Conferences on Artificial Intelligence (IJCAI)*, pages 348–353.
- [Euzenat et al., 2007] Euzenat, J., Isaac, A., Meilicke, C., Shvaiko, P., Stuckenschmidt, H., Svab, O., Svatek, V., van Hage, W. R., and Yatskevich, M. (2007). First results of the ontology alignment evaluation initiative 2007. In *Ontology Matching Workshop at International Semantic Web Conference (ISWC)*.
- [Euzenat et al., 2004] Euzenat, J., Le Bach, T., Paolo, Debo, J., Kuntz, R. D., Ehrig, M., Hauswirth, M., Jarrar, M., Lara, R., Dianamaynard, Napoli, A., Stamou, G., Stuckenschmidt, H., Shvaiko, P., Sergiotessaris, Van Acker, S., and Zaihrayeu, I. (2004). State of the art on ontology deliverable d2.2.3.
- [Euzenat et al., 2006] Euzenat, J., Mochol, M., Shvaiko, P., Stuckenschmidt, H., Svab, O., Svatek, V., van Hage, W. R., and Yatskevich, M. (2006). Results of the ontology alignment evaluation initiative 2006. In *Ontology Matching Workshop at International Semantic Web Conference (ISWC)*.
- [Euzenat and Shvaiko, 2007] Euzenat, J. and Shvaiko, P. (2007). *Ontology Matching*. Springer-Verlag New York, Inc. Secaucus, NJ, USA.
- [Euzenat et al., 2005] Euzenat, J., Stuckenschmidt, H., and Yatskevich, M. (2005). Introduction to the ontology alignment evaluation 2005. In *Integrating Ontologies*.
- [Fleiss et al., 1981] Fleiss, J. et al. (1981). *Statistical Methods for Rates and Proportions*. John Wiley & Sons New York, N. Y.
- [Fung and Yee, 1998] Fung, P. and Yee, L. Y. (1998). An IR approach for translating new words from nonparallel, comparable texts. In Boitet, C. and Whitelock, P., editors, *Proceedings of the Thirty-Sixth Annual Meeting of the Association for Computational Linguistics and Seventeenth International Conference on Computational Linguistics*, pages 414–420, San Francisco, California. Morgan Kaufmann Publishers.

- [Giunchiglia et al., 2004] Giunchiglia, F., Shvaiko, P., and Yatskevich, M. (2004). S-match: an algorithm and an implementation of semantic matching. In *Proceedings of the European Semantic Web Symposium (ESWC)*, pages 61–75.
- [Giunchiglia et al., 2006] Giunchiglia, F., Shvaiko, P., and Yatskevich, M. (2006). Discovering missing background knowledge in ontology matching. In *ECAI*, pages 382–386.
- [Gligorov et al., 2007] Gligorov, R., Aleksovski, Z., ten Kate, W., and van Harmelen, F. (2007). Using google distance to weight approximate ontology matches. In *Proc. of World Wide Web Conf.*, pages 767–776.
- [Gruber, 1993] Gruber, T. R. (1993). Toward principles for the design of ontologies used for knowledge sharing. *Int. J. Hum.-Comput. Stud.*, 43(5-6):907–928.
- [Guha and McCool, 2003] Guha, R. and McCool, R. (2003). Tap: A semantic web toolkit. *Semantic Web Journal*, 1(1):81–87.
- [Hammer, 1998] Hammer, E. M. (1998). Semantics for existential graphs. *Journal of Philosophical Logic*, 27(5):489–503.
- [Hamming, 1950] Hamming, R. W. (1950). Error detecting and error correcting codes. *Bell System Technical Journal*, 26(2):147–160.
- [Hearst, 1992] Hearst, M. A. (1992). Automatic acquisition of hyponyms from large text corpora. In *Proceedings of the 14th conference on Computational linguistics*, pages 539–545.
- [Horrocks et al., 2003] Horrocks, I., Patel-Schneider, P. F., and van Harmelen, F. (2003). From shiq and rdf to owl: the making of a web ontology language. *Journal of Web Semantics*, 1(1):7–26.
- [Hu et al., 2005] Hu, W., Jian, N., Qu, Y., and Wang, Y. (2005). Gmo: A graph matching for ontologies. In *Integrating Ontologies*.
- [Hull, 1996] Hull, D. A. (1996). Stemming algorithms: A case study for detailed evaluation. *Journal of the American Society of Information Science*, 47(1):70–84.
- [H.Wache et al., 2001] H.Wache, V’ogele, T., Visser, U., Stuckenschmidt, H., Schuster, G., Neumann, H., and H’ubner, S. (2001). Ontology-based integration of information - a survey of existing approaches. In *Proceedings of*



*the workshop on Ontologies and Information Sharing at the International Joint Conference on Artificial Intelligence (IJCAI)*, pages 108–117.

- [Ichise et al., 2003] Ichise, R., Takeda, H., and Honiden, S. (2003). Integrating multiple internet directories by instance-based learning. In *IJCAI*, pages 22–30.
- [Jaro, 1995] Jaro, M. A. (1995). Probabilistic linkage of large public health data file. In *Statistics in Medicine*, volume 14, pages 491–498.
- [Jian et al., 2005] Jian, N., Hu, W., Cheng, G., and Qu, Y. (2005). Falconao: Aligning ontologies with falcon. In *K-Cap 2005 Workshop on Integrating Ontologies*.
- [Lassila et al., 1999] Lassila, O., Swick, R., et al. (1999). Resource description framework (rdf) model and syntax specification.
- [Lenat, 1995] Lenat, D. (1995). Cyc: a large-scale investment in knowledge infrastructure. *Communications of the ACM*, 38(11):33–38.
- [Levenshtein, 1966] Levenshtein, V. I. (1966). Binary codes capable of correcting deletions, insertions, and reversals. *Soviet Physics Doklady*, 10(8):707–710.
- [Lipscomb, 2000] Lipscomb, C. (2000). Medical subject headings (mesh). *Bull Med Libr Assoc*, 88(3):265–266.
- [Liu et al., 2004] Liu, S., Liu, F., Yu, C., and Meng, W. (2004). An effective approach to document retrieval via utilizing wordnet and recognizing phrases. In *Proceedings of the 27th annual International Conference on Research and development in information retrieval (ACM SIGIR)*, pages 266–272, New York, NY, USA. ACM Press.
- [Lowe and Barnett, 1994] Lowe, H. and Barnett, G. (1994). Understanding and using the medical subject headings (mesh) vocabulary to perform literature searches. *JAMA*, 271(14):1103–1108.
- [Madhavan et al., 2001a] Madhavan, J., Bernstein, P. A., and Rahm, E. (2001a). Generic schema matching with cupid. In *Proceedings of the 27th VLDB Conference, Roma, Italy, 2001*.
- [Madhavan et al., 2001b] Madhavan, J., Bernstein, P. A., and Rahm, E. (2001b). Generic schema matching with cupid. In *VLDB '01: Proceedings of the 27th International Conference on Very Large Data Bases*, pages 49–58, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

- [Magnini et al., 2004] Magnini, B., Speranza, M., and Girardi, C. (2004). A semantic-based approach to interoperability of classification hierarchies: evaluation of linguistic techniques. In *COLING '04: Proceedings of the 20th international conference on Computational Linguistics*, page 1133, Morristown, NJ, USA. Association for Computational Linguistics.
- [McCallum et al., 2000] McCallum, A., Nigam, K., and Ungar, L. H. (2000). Efficient clustering of high-dimensional data sets with application to reference matching. In *KDD '00: Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 169–178, New York, NY, USA. ACM Press.
- [Meilicke et al., 2006] Meilicke, C., Stuckenschmidt, H., and Tamin, A. (2006). Improving automatically created mappings using logical reasoning. In *Ontology Matching Workshop at International Semantic Web Conference (ISWC)*.
- [Minsky, 1985] Minsky, M. (1985). A framework for representing knowledge. In Brachman, R. J. and Levesque, H. J., editors, *Readings in Knowledge Representation*, pages 245–262. Kaufmann, Los Altos, CA.
- [Mork et al., 2004] Mork, P., Pottinger, R. A., and Bernstein, P. A. (2004). Challenges in precisely aligning models of human anatomy using generic schema matching. Technical report, Microsoft Research.
- [Motik et al., 2007] Motik, B., Horrocks, I., and Sattler, U. (2007). Bridging the gap between owl and relational databases. In *WWW '07: Proceedings of the 16th international conference on World Wide Web*, pages 807–816, New York, NY, USA. ACM Press.
- [Needleman and Wunsch, 1970] Needleman, S. and Wunsch, C. (1970). A general method applicable to the search for similarities in the amino acid sequence of two proteins. *J Mol Biol*, 48(3):443–53.
- [Noy and Musen, 2000] Noy, N. F. and Musen, M. A. (2000). PROMPT: Algorithm and Tool for Automated Ontology Merging and Alignment. In *Proceedings of the Seventeenth National Conference on Artificial Intelligence (AAAI-2000)*, Austin, TX. AAAI/MIT Press.
- [Pachet and Cazaly, 2000] Pachet, F. and Cazaly, D. (2000). A taxonomy of musical genres. In *RIAO'00: Proc. Content-Based Multimedia Information Access*.

- [Quillian, 1985] Quillian, M. R. (1985). Word concepts: A theory and simulation of some basic semantic capabilities. In Brachman, R. J. and Levesque, H. J., editors, *Readings in Knowledge Representation*, pages 97–118. Kaufmann, Los Altos, CA.
- [Rahm and Bernstein, 2001] Rahm, E. and Bernstein, P. A. (2001). A Survey of Approaches to Automatic Schema Matching. *VLDB Journal*, 10(4).
- [Resnik, 1999] Resnik, P. (1999). Semantic similarity in a taxonomy: An information-based measure and its application to problems of ambiguity in natural language. *Journal of Artificial Intelligence Research*, 11:95–130.
- [Roddick et al., 2003] Roddick, J. F., Hornsby, K., and de Vries, D. (2003). A unifying semantic distance model for determining the similarity of attribute values. In *ACSC '03: Proceedings of the 26th Australasian computer science conference*, pages 111–118, Darlinghurst, Australia, Australia. Australian Computer Society, Inc.
- [Rose and Jr., 2003] Rose, C. and Jr., J. L. V. M. (2003). Reference ontology for biomedical informatics: the foundational model of anatomy. *Journal of biomedical informatics*, 36:478–500.
- [Sabou et al., 2006] Sabou, M., dAquin, M., and Motta, E. (2006). Using the semantic web as background knowledge for ontology mapping. In *Proceedings of Ontology Matching Workshop*.
- [Schlobach, 2004] Schlobach, S. (2004). Semantic clarification by pinpointing. In *Proceedings of the second European Semantic Web conference*. LNCS. Springer Verlag.
- [Schlobach et al., 2007] Schlobach, S., Klein, M., and Peelen, L. (2007). Description logics with approximate definitions: Precise modeling of vague concepts. In Veloso, M., editor, *Proceedings of the 20th International Joint Conference on Artificial Intelligence, IJCAI 07*, Hyderabad, India.
- [Shvaiko and Euzenat, 2005] Shvaiko, P. and Euzenat, J. (2005). A survey of schema-based matching approaches. *Journal on Data Semantics*, 4:146–171.
- [Smith and Waterman, 1981] Smith, T. F. and Waterman, M. S. (1981). Identification of common molecular subsequences. *Journal of Molecular Biology*, 147:195–197.

- [Sowa, 2000] Sowa, J. (2000). *Knowledge Representation: Logical, Philosophical, and Computational Foundations*. MIT Press.
- [Sowa, 1979] Sowa, J. F. (1979). Semantics of conceptual graphs. In *Proceedings of the 17th annual meeting on Association for Computational Linguistics*, pages 39–44, Morristown, NJ, USA. Association for Computational Linguistics.
- [Straccia, 2006] Straccia, U. (2006). A fuzzy description logic for the semantic web. In Sanchez, E., editor, *Fuzzy Logic and the Semantic Web, Capturing Intelligence*, chapter 4, pages 73–90. Elsevier.
- [Valtchev and Euzenat, 1997] Valtchev, P. and Euzenat, J. (1997). Dissimilarity measure for collections of objects and values. In *IDA '97: Proceedings of the Second International Symposium on Advances in Intelligent Data Analysis, Reasoning about Data*, pages 259–272, London, UK. Springer-Verlag.
- [van Hage et al., 2006] van Hage, W. R., Kolb, H., and Schreiber, G. (2006). A method for learning part-whole relations. In *International Semantic Web Conference*, pages 723–735.
- [Van Rijsbergen, 1979] Van Rijsbergen, C. (1979). *Information Retrieval*. Butterworth-Heinemann Newton, MA, USA.
- [Wang et al., 2006] Wang, T. D., Parsia, B., and Hendler, J. A. (2006). A survey of the web ontology landscape. In Cruz, I. F., Decker, S., Allemang, D., Preist, C., Schwabe, D., Mika, P., Uschold, M., and Aroyo, L., editors, *International Semantic Web Conference*, volume 4273 of *Lecture Notes in Computer Science*, pages 682–694. Springer.
- [Winkler, 1999] Winkler, W. E. (1999). The state of record linkage and current research problems. Technical report, Statistical Research Division, U.S. Bureau of the Census.
- [Woods, 1985] Woods, W. A. (1985). What’s in a link: Foundations for semantic networks. In Brachman, R. J. and Levesque, H. J., editors, *Readings in Knowledge Representation*, pages 217–241. Kaufmann, Los Altos, CA.
- [Zhang and Bodenreider, 2003] Zhang, S. and Bodenreider, O. (2003). Aligning representations of anatomy using lexical and structural methods. *AMIA Symposium*, pages 753–757.

- [Zhang and Bodenreider, 2005] Zhang, S. and Bodenreider, O. (2005). Alignment of multiple ontologies of anatomy: deriving indirect mappings from direct mappings to a reference. In *Proceedings of AMIA Annual Symposium*, pages 864–868, U. S. National Library of Medicine, Bethesda, Maryland, National Institutes of Health, Department of Health & Human Services, USA. smhang@math.ac.cn.

# SIKS Dissertatiereeks

## 1998

- 1998-01** Johan van den Akker (CWI)  
DEGAS - An Active, Temporal Database of Autonomous Objects
- 1998-02** Floris Wiesman (UM)  
Information Retrieval by Graphically Browsing Meta-Information
- 1998-03** Ans Steuten (TUD)  
A Contribution to the Linguistic Analysis of Business Conversations  
within the Language/Action Perspective
- 1998-04** Dennis Breuker (UM)  
Memory versus Search in Games
- 1998-05** E.W.Oskamp (RUL)  
Computerondersteuning bij Straftoemeting

## 1999

- 1999-01** Mark Sloof (VU)  
Physiology of Quality Change Modelling; Automated modelling of  
Quality Change of Agricultural Products
- 1999-02** Rob Potharst (EUR)  
Classification using decision trees and neural nets
- 1999-03** Don Beal (UM)  
The Nature of Minimax Search
- 1999-04** Jacques Penders (UM)  
The practical Art of Moving Physical Objects

- 1999-05** Aldo de Moor (KUB)  
Empowering Communities: A Method for the Legitimate  
User-Driven Specification of Network Information Systems
- 1999-06** Niek J.E. Wijngaards (VU)  
Re-design of compositional systems
- 1999-07** David Spelt (UT)  
Verification support for object database design
- 1999-08** Jacques H.J. Lenting (UM)  
Informed Gambling: Conception and Analysis of  
a Multi-Agent Mechanism for Discrete Reallocation.

## 2000

- 2000-01** Frank Niessink (VU)  
Perspectives on Improving Software Maintenance
- 2000-02** Koen Holtman (TUE)  
Prototyping of CMS Storage Management
- 2000-03** Carolien M.T. Metselaar (UVA)  
Sociaal-organisatorische gevolgen van kennistechnologie;  
een procesbenadering en actorperspectief.
- 2000-04** Geert de Haan (VU)  
ETAG, A Formal Model of Competence Knowledge for  
User Interface Design
- 2000-05** Ruud van der Pol (UM)  
Knowledge-based Query Formulation in Information Retrieval.
- 2000-06** Rogier van Eijk (UU)  
Programming Languages for Agent Communication
- 2000-07** Niels Peek (UU)  
Decision-theoretic Planning of Clinical Patient Management
- 2000-08** Veerle Coup (EUR)  
Sensitivity Analysis of Decision-Theoretic Networks
- 2000-09** Florian Waas (CWI)  
Principles of Probabilistic Query Optimization

- 2000-10** Niels Nes (CWI)  
Image Database Management System Design Considerations,  
Algorithms and Architecture
- 2000-11** Jonas Karlsson (CWI)  
Scalable Distributed Data Structures for Database Management

## 2001

- 2001-01** Silja Renooij (UU)  
Qualitative Approaches to Quantifying Probabilistic Networks
- 2001-02** Koen Hindriks (UU)  
Agent Programming Languages: Programming with Mental Models
- 2001-03** Maarten van Someren (UvA)  
Learning as problem solving
- 2001-04** Evgueni Smirnov (UM)  
Conjunctive and Disjunctive Version Spaces with  
Instance-Based Boundary Sets
- 2001-05** Jacco van Ossenbruggen (VU)  
Processing Structured Hypermedia: A Matter of Style
- 2001-06** Martijn van Welie (VU)  
Task-based User Interface Design
- 2001-07** Bastiaan Schonhage (VU)  
Diva: Architectural Perspectives on Information Visualization
- 2001-08** Pascal van Eck (VU)  
A Compositional Semantic Structure for Multi-Agent Systems Dynamics.
- 2001-09** Pieter Jan 't Hoen (RUL)  
Towards Distributed Development of Large Object-Oriented Models,  
Views of Packages as Classes
- 2001-10** Maarten Sierhuis (UvA)  
Modeling and Simulating Work Practice BRAHMS:  
a multiagent modeling and simulation language for  
work practice analysis and design
- 2001-11** Tom M. van Engers (VUA)  
Knowledge Management:  
The Role of Mental Models in Business Systems Design



**2002**

- 2002-01** Nico Lassing (VU)  
Architecture-Level Modifiability Analysis
- 2002-02** Roelof van Zwol (UT)  
Modelling and searching web-based document collections
- 2002-03** Henk Ernst Blok (UT)  
Database Optimization Aspects for Information Retrieval
- 2002-04** Juan Roberto Castelo Valdueza (UU)  
The Discrete Acyclic Digraph Markov Model in Data Mining
- 2002-05** Radu Serban (VU)  
The Private Cyberspace Modeling Electronic  
Environments inhabited by Privacy-concerned Agents
- 2002-06** Laurens Mommers (UL)  
Applied legal epistemology:  
Building a knowledge-based ontology of the legal domain
- 2002-07** Peter Boncz (CWI)  
Monet: A Next-Generation DBMS Kernel For  
Query-Intensive Applications
- 2002-08** Jaap Gordijn (VU)  
Value Based Requirements Engineering: Exploring Innovative  
E-Commerce Ideas
- 2002-09** Willem-Jan van den Heuvel (KUB)  
Integrating Modern Business Applications with  
Objectified Legacy Systems
- 2002-10** Brian Sheppard (UM)  
Towards Perfect Play of Scrabble
- 2002-11** Wouter C.A. Wijngaards (VU)  
Agent Based Modelling of Dynamics  
Biological and Organisational Applications
- 2002-12** Albrecht Schmidt (Uva)  
Processing XML in Database Systems
- 2002-13** Hongjing Wu (TUE)  
A Reference Architecture for Adaptive Hypermedia Applications
- 2002-14** Wieke de Vries (UU)  
Agent Interaction: Abstract Approaches to Modelling,  
Programming and Verifying Multi-Agent Systems

- 2002-15** Rik Eshuis (UT)  
Semantics and Verification of UML Activity Diagrams  
for Workflow Modelling
- 2002-16** Pieter van Langen (VU)  
The Anatomy of Design: Foundations, Models and Applications
- 2002-17** Stefan Manegold (UVA)  
Understanding, Modeling, and Improving  
Main-Memory Database Performance

## 2003

- 2003-01** Heiner Stuckenschmidt (VU)  
Ontology-Based Information Sharing in Weakly Structured Environments
- 2003-02** Jan Broersen (VU)  
Modal Action Logics for Reasoning About Reactive Systems
- 2003-03** Martijn Schuemie (TUD)  
Human-Computer Interaction and  
Presence in Virtual Reality Exposure Therapy
- 2003-04** Milan Petkovic (UT)  
Content-Based Video Retrieval Supported by Database Technology
- 2003-05** Jos Lehmann (UVA)  
Causation in Artificial Intelligence and Law - A modelling approach
- 2003-06** Boris van Schooten (UT)  
Development and specification of virtual environments
- 2003-07** Machiel Jansen (UvA)  
Formal Explorations of Knowledge Intensive Tasks
- 2003-08** Yongping Ran (UM)  
Repair Based Scheduling
- 2003-09** Rens Kortmann (UM)  
The resolution of visually guided behaviour
- 2003-10** Andreas Lincke (UvT)  
Electronic Business Negotiation: Some experimental studies on  
the interaction between medium, innovation context and culture

- 2003-11** Simon Keizer (UT)  
Reasoning under Uncertainty in Natural Language Dialogue  
using Bayesian Networks
- 2003-12** Roeland Ordelman (UT)  
Dutch speech recognition in multimedia information retrieval
- 2003-13** Jeroen Donkers (UM)  
Nosce Hostem - Searching with Opponent Models
- 2003-14** Stijn Hoppenbrouwers (KUN)  
Freezing Language: Conceptualisation Processes  
across ICT-Supported Organisations
- 2003-15** Mathijs de Weerd (TUD)  
Plan Merging in Multi-Agent Systems
- 2003-16** Menzo Windhouwer (CWI)  
Feature Grammar Systems - Incremental Maintenance  
of Indexes to Digital Media Warehouses
- 2003-17** David Jansen (UT)  
Extensions of Statecharts with Probability,  
Time, and Stochastic Timing
- 2003-18** Levente Kocsis (UM)  
Learning Search Decisions

## 2004

- 2004-01** Virginia Dignum (UU)  
A Model for Organizational Interaction:  
Based on Agents, Founded in Logic
- 2004-02** Lai Xu (UvT)  
Monitoring Multi-party Contracts for E-business
- 2004-03** Perry Groot (VU)  
A Theoretical and Empirical Analysis  
of Approximation in Symbolic Problem Solving
- 2004-04** Chris van Aart (UVA)  
Organizational Principles for Multi-Agent Architectures
- 2004-05** Viara Popova (EUR)  
Knowledge discovery and monotonicity

- 2004-06** Bart-Jan Hommes (TUD)  
The Evaluation of Business Process Modeling Techniques
- 2004-07** Elise Boltjes (UM)  
Voorbeeldig onderwijs; voorbeeldgestuurd onderwijs,  
een opstap naar abstract denken, vooral voor meisjes
- 2004-08** Joop Verbeek(UM)  
Politie en de Nieuwe Internationale Informatiemarkt,  
Grensregionale politile gegevensuitwisseling en digitale expertise
- 2004-09** Martin Caminada (VU)  
For the Sake of the Argument;  
explorations into argument-based reasoning
- 2004-10** Suzanne Kabel (UVA)  
Knowledge-rich indexing of learning-objects
- 2004-11** Michel Klein (VU)  
Change Management for Distributed Ontologies
- 2004-12** The Duy Bui (UT)  
Creating emotions and facial expressions for embodied agents
- 2004-13** Wojciech Jamroga (UT)  
Using Multiple Models of Reality: On Agents who Know how to Play
- 2004-14** Paul Harrenstein (UU)  
Logic in Conflict. Logical Explorations in Strategic Equilibrium
- 2004-15** Arno Knobbe (UU)  
Multi-Relational Data Mining
- 2004-16** Federico Divina (VU)  
Hybrid Genetic Relational Search for Inductive Learning
- 2004-17** Mark Winands (UM)  
Informed Search in Complex Games
- 2004-18** Vania Bessa Machado (UvA)  
Supporting the Construction of Qualitative Knowledge Models
- 2004-19** Thijs Westerveld (UT)  
Using generative probabilistic models for multimedia retrieval
- 2004-20** Madelon Evers (Nyenrode)  
Learning from Design: facilitating multidisciplinary design teams

**2005**

- 2005-01** Floor Verdenius (UVA)  
Methodological Aspects of Designing Induction-Based Applications
- 2005-02** Erik van der Werf (UM)  
AI techniques for the game of Go
- 2005-03** Franc Grootjen (RUN)  
A Pragmatic Approach to the Conceptualisation of Language
- 2005-04** Nirvana Meratnia (UT)  
Towards Database Support for Moving Object data
- 2005-05** Gabriel Infante-Lopez (UVA)  
Two-Level Probabilistic Grammars for Natural Language Parsing
- 2005-06** Pieter Spronck (UM)  
Adaptive Game AI
- 2005-07** Flavius Frasincar (TUE)  
Hypermedia Presentation Generation for  
Semantic Web Information Systems
- 2005-08** Richard Vdovjak (TUE)  
A Model-driven Approach for Building  
Distributed Ontology-based Web Applications
- 2005-09** Jeen Broekstra (VU)  
Storage, Querying and Inferencing for Semantic Web Languages
- 2005-10** Anders Bouwer (UVA)  
Explaining Behaviour: Using Qualitative Simulation in  
Interactive Learning Environments
- 2005-11** Elth Ogston (VU)  
Agent Based Matchmaking and Clustering -  
A Decentralized Approach to Search
- 2005-12** Csaba Boer (EUR)  
Distributed Simulation in Industry
- 2005-13** Fred Hamburg (UL)  
Een Computermodeel voor het Ondersteunen  
van Euthanasiebeslissingen
- 2005-14** Borys Omelayenko (VU)  
Web-Service configuration on the Semantic Web;  
Exploring how semantics meets pragmatics

- 2005-15** Tibor Bosse (VU)  
Analysis of the Dynamics of Cognitive Processes
- 2005-16** Joris Graaumans (UU)  
Usability of XML Query Languages
- 2005-17** Boris Shishkov (TUD)  
Software Specification Based on Re-usable Business Components
- 2005-18** Danielle Sent (UU)  
Test-selection strategies for probabilistic networks
- 2005-19** Michel van Dartel (UM)  
Situated Representation
- 2005-20** Cristina Coteanu (UL)  
Cyber Consumer Law, State of the Art and Perspectives
- 2005-21** Wijnand Derks (UT)  
Improving Concurrency and Recovery in Database Systems  
by Exploiting Application Semantics

## 2006

- 2006-01** Samuil Angelov (TUE)  
Foundations of B2B Electronic Contracting
- 2006-02** Cristina Chisalita (VU)  
Contextual issues in the design and  
use of information technology in organizations
- 2006-03** Noor Christoph (UVA)  
The role of metacognitive skills in learning to solve problems
- 2006-04** Marta Sabou (VU)  
Building Web Service Ontologies
- 2006-05** Cees Pierik (UU)  
Validation Techniques for Object-Oriented Proof Outlines
- 2006-06** Ziv Baida (VU)  
Software-aided Service Bundling -  
Intelligent Methods & Tools for Graphical Service Modeling
- 2006-07** Marko Smiljanic (UT)  
XML schema matching – balancing efficiency and  
effectiveness by means of clustering

- 2006-08** Eelco Herder (UT)  
Forward, Back and Home Again - Analyzing  
User Behavior on the Web
- 2006-09** Mohamed Wahdan (UM)  
Automatic Formulation of the Auditor's Opinion
- 2006-10** Ronny Siebes (VU)  
Semantic Routing in Peer-to-Peer Systems
- 2006-11** Joeri van Ruth (UT)  
Flattening Queries over Nested Data Types
- 2006-12** Bert Bongers (VU)  
Interactivation - Towards an e-cology of people,  
our technological environment, and the arts
- 2006-13** Henk-Jan Lebbink (UU)  
Dialogue and Decision Games for Information Exchanging Agents
- 2006-14** Johan Hoorn (VU)  
Software Requirements: Update, Upgrade, Redesign -  
towards a Theory of Requirements Change
- 2006-15** Rainer Malik (UU)  
CONAN: Text Mining in the Biomedical Domain
- 2006-16** Carsten Riggelsen (UU)  
Approximation Methods for Efficient Learning  
of Bayesian Networks
- 2006-17** Stacey Nagata (UU)  
User Assistance for Multitasking with  
Interruptions on a Mobile Device
- 2006-18** Valentin Zhizhkun (UVA)  
Graph transformation for Natural Language Processing
- 2006-19** Birna van Riemsdijk (UU)  
Cognitive Agent Programming: A Semantic Approach
- 2006-20** Marina Velikova (UvT)  
Monotone models for prediction in data mining
- 2006-21** Bas van Gils (RUN)  
Aptness on the Web
- 2006-22** Paul de Vrieze (RUN)  
Fundamentals of Adaptive Personalisation

- 2006-23** Ion Juvina (UU)  
Development of Cognitive Model for Navigating on the Web
- 2006-24** Laura Hollink (VU)  
Semantic Annotation for Retrieval of Visual Resources
- 2006-25** Madalina Drugan (UU)  
Conditional log-likelihood MDL and Evolutionary MCMC
- 2006-26** Vojkan Mihajlovic (UT)  
Score Region Algebra: A Flexible Framework for  
Structured Information Retrieval
- 2006-27** Stefano Bocconi (CWI)  
Vox Populi: generating video documentaries from  
semantically annotated media repositories
- 2006-28** Borkur Sigurbjornsson (UVA)  
Focused Information Access using XML Element Retrieval

## 2007

- 2007-01** Kees Leune (UvT)  
Access Control and Service-Oriented Architectures
- 2007-02** Wouter Teepe (RUG)  
Reconciling Information Exchange and Confidentiality:  
A Formal Approach
- 2007-03** Peter Mika (VU)  
Social Networks and the Semantic Web
- 2007-04** Jurriaan van Diggelen (UU)  
Achieving Semantic Interoperability in Multi-agent Systems:  
a dialogue-based approach
- 2007-05** Bart Schermer (UL)  
Software Agents, Surveillance, and the Right to Privacy:  
a Legislative Framework for Agent-enabled Surveillance
- 2007-06** Gilad Mishne (UVA)  
Applied Text Analytics for Blogs
- 2007-07** Natasa Jovanovic' (UT)  
To Whom It May Concern - Addressee Identification in  
Face-to-Face Meetings



- 2007-08** Mark Hoogendoorn (VU)  
Modeling of Change in Multi-Agent Organizations
- 2007-09** David Mobach (VU)  
Agent-Based Mediated Service Negotiation
- 2007-10** Huib Aldewereld (UU)  
Autonomy vs. Conformity: an Institutional Perspective on  
Norms and Protocols
- 2007-11** Natalia Stash (TUE)  
Incorporating Cognitive/Learning Styles in a General-Purpose  
Adaptive Hypermedia System
- 2007-12** Marcel van Gerven (RUN)  
Bayesian Networks for Clinical  
Decision Support: A Rational Approach to  
Dynamic Decision-Making under Uncertainty
- 2007-13** Rutger Rienks (UT)  
Meetings in Smart Environments;  
Implications of Progressing Technology
- 2007-14** Niek Bergboer (UM)  
Context-Based Image Analysis
- 2007-15** Joyca Lacroix (UM)  
NIM: a Situated Computational Memory Model
- 2007-16** Davide Grossi (UU)  
Designing Invisible Handcuffs. Formal investigations in  
Institutions and Organizations for Multi-agent Systems
- 2007-17** Theodore Charitos (UU)  
Reasoning with Dynamic Networks in Practice
- 2007-18** Bart Orriens (UvT)  
On the development an management of adaptive  
business collaborations
- 2007-19** David Levy (UM)  
Intimate relationships with artificial partners
- 2007-20** Slinger Jansen (UU)  
Customer Configuration Updating in a Software Supply Network
- 2007-21** Karianne Vermaas (UU)  
Fast diffusion and broadening use: A research on  
residential adoption and usage of broadband internet in  
the Netherlands between 2001 and 2005

- 2007-22** Zlatko Zlatev (UT)  
Goal-oriented design of value and process models from patterns
- 2007-23** Peter Barna (TUE)  
Specification of Application Logic in Web Information Systems
- 2007-24** Georgina Ramirez Camps (CWI)  
Structural Features in XML Retrieval
- 2007-25** Joost Schalken (VU)  
Empirical Investigations in Software Process Improvement

## 2008

- 2008-01** Katalin Boer-Sorban (EUR)  
Agent-Based Simulation of Financial Markets: A modular, continuous-time approach
- 2008-02** Alexei Sharpanskykh (VU)  
On Computer-Aided Methods for Modeling and Analysis of Organizations
- 2008-03** Vera Hollink (UVA)  
Optimizing hierarchical menus: a usage-based approach
- 2008-04** Ander de Keijzer (UT)  
Management of Uncertain Data - towards unattended integration
- 2008-05** Bela Mutschler (UT)  
Modeling and simulating causal dependencies on process-aware information systems from a cost perspective
- 2008-06** Arjen Hommersom (RUN)  
On the Application of Formal Methods to Clinical Guidelines, an Artificial Intelligence Perspective
- 2008-07** Peter van Rosmalen (OU)  
Supporting the tutor in the design and support of adaptive e-learning
- 2008-08** Janneke Bolt (UU)  
Bayesian Networks: Aspects of Approximate Inference
- 2008-09** Christof van Nimwegen (UU)  
The paradox of the guided user: assistance can be counter-effective
- 2008-10** Wauter Bosma (UT)  
Discourse oriented summarization

- 2008-11** Vera Kartseva (VU)  
Designing Controls for Network Organizations:  
A Value-Based Approach
- 2008-12** Jozsef Farkas (RUN)  
A Semiotically Oriented Cognitive Model of  
Knowledge Representation
- 2008-13** Caterina Carraciolo (UVA)  
Topic Driven Access to Scientific Handbooks
- 2008-14** Arthur van Bunningen (UT)  
Context-Aware Querying; Better Answers with Less Effort
- 2008-15** Martijn van Otterlo (UT)  
The Logic of Adaptive Behavior: Knowledge Representation  
and Algorithms for the Markov Decision  
Process Framework in First-Order Domains.
- 2008-16** Henriette van Vugt (VU)  
Embodied agents from a user's perspective
- 2008-17** Martin Op 't Land (TUD)  
Applying Architecture and Ontology to the Splitting and  
Allying of Enterprises
- 2008-18** Guido de Croon (UM)  
Adaptive Active Vision
- 2008-19** Henning Rode (UT)  
From Document to Entity Retrieval: Improving Precision and  
Performance of Focused Text Search
- 2008-20** Rex Arendsen (UVA)  
Geen bericht, goed bericht. Een onderzoek naar de effecten  
van de introductie van elektronisch berichtenverkeer met  
de overheid op de administratieve lasten van bedrijven.
- 2008-21** Krisztian Balog (UVA)  
People Search in the Enterprise
- 2008-22** Henk Koning (UU)  
Communication of IT-Architecture
- 2008-23** Stefan Visscher (UU)  
Bayesian network models for the management of  
ventilator-associated pneumonia