Ontology mapping with auxiliary resources

Ontology Mapping with Auxiliary Resources

Frederik Christiaan Schadd

Version 25. Oktober 2015

Ontology Mapping with Auxiliary Resources

Ontology Mapping with Auxiliary Resources

PROEFSCHRIFT

ter verkrijging van de graad van doctor
aan de Universiteit Maastricht,
op gezag van de Rector Magnificus,
Prof. dr. L.L.G. Soete,
volgens het besluit van het College van Decanen,
in het openbaar te verdedigen
op donderdag 17 december 2015 om 14.00 uur

door

Frederik Christiaan Schadd

Promotor: Prof. dr. ir. J.C. Scholtes
Copromotor: Dr. ir. ing. N. Roos

Leden van de beoordelingscommissie:
Prof. dr. ir. R.L.M. Peeters (*chair*)
Prof. dr. C.T.H. Evelo
Prof. dr. ir. F.A.H. van Harmelen (VU University Amsterdam)
Prof. dr. H. Stuckenschmidt (University of Mannheim)
Prof. dr. G.B. Weiss

This research has been funded by the transnational University of Limburg (tUL).

Dissertation Series No. 2015-33
The research reported in this thesis has been carried out under the auspices of SIKS, the Dutch Research School for Information and Knowledge Systems.

Printed by *Datawyze, Maastricht*
ISBN 978-94-6159-499-0

Preface

First and foremost, I'd like to thank my supervisor Nico Roos for his continuous support, stimulation and encouragement over the past years. It is due to his guidance that was it possible to channel my efforts into several publications and ultimately this thesis. I have thoroughly enjoyed our thoughtful, prompt and open discussions about research, teaching or any current topics. The humorous nature of our discussions made our weekly meeting always a joy, never a chore. He gave me the freedom to pursue my research ideas and actively shape this project. I would also like to thank Jan Scholtes agreeing to be my promotor and his insightful feedback, which helped in shaping this thesis into what it is today.

I'd also to thank my colleagues and friends at DKE, specifically: Daan Bloembergen, Hendrik Baier, Steven de Jong, Nyree Lemmens, Philippe Uyttendaele, Nela Lekic, Matthijs Cluitmans, Daniel Claes, Haitham Bou Ammar, Michael Kaisers, Daniel Hennes, Michael Clerx, Pietro Bonizzi, Marc Lanctot, Pim Nijssen, Siqi Chen, Wei Zhao, Zhenlong Sun, Nasser Davarzani and Bijan Ranjbarsahraei. Sharing stories during lunch time and creating new ones at conferences, courses or pubs has been a highlight during my time as a PhD.

Further, I'd like to thank the frequent participants of the PhD-Academy events, such as drinks, movie-nights or outdoor activities. I'd like to thank the following people who made my stay in Maastricht particularly fun and exciting: Julie Dela Cruz, Gabri Marconi, Mare Oehlen, Mehrdad Seirafi, Sanne ten Oever, Joost Mulders, Shuan Ghazi, Tia Ammerlaan, Rina Tsubaki, Howard Hudson, Lisa Bushart, Anna Zseleva, Annelore Verhagen, Burcu Duygu, Christine Gutekunst, Paola Spataro, Anne-Sophie Warda, Masaoki Ishiguro, Barbara Zarzycka, Paula Nagler, Nordin Hanssen, Anna Zseleva, Roxanne Korthals, Zubin Vincent, Łukasz Wojtulewicz, Paola Spataro, Peter Barton, Bas Ganzevles, Jo-Anne Illidge, Dorijn Hertroijs, Eveline van Velthuijsen, Hedda Munthe, Mahdi Al Taher, Ibrahima Sory Kaba, Mueid Ayub and Jessie Lemmens . During my time as a PhD I also started climbing as a new hobby, thanks to which I met the following great people: Nico Salamanca, Jan Feld, Frauke Meyer, Maria Zumbühl and Martijn Moorlag. Thanks to you I always had something to look forward to, even if things weren't going well with research, be it going bouldering every week, climbing outside in France or Belgium, or any non-sporty activity with the other great PhD-Academy people. Further, I'd like to thank the members of MaasSAC, the Maastricht student climbing association, for the many fun evenings of either climbing or other social activities.

Wie heisst es doch so schön? Ob Norden, Süden, Osten oder Westen, zu Hause *ist es doch am besten. Ich bin dankbar für meine Freunde in meiner Heimat, besonders Marcel Ludwig, Markus Bock, Sebastian Müller, Christine Müller, Alexander Miesen, Christopher Löltgen, Daniel Adenau, Dominik Fischer, Stefan Kolb, Stefan Schiffer, Michael Lichtner, Stefan Bauman und Andreas Bauman für die ganzen tollen Jahre die wir zusammen verbracht haben. Auf euch konnte ich mich immer verlassen um Spass zu haben und um meinen Verstand wieder zu gewinnen, wann auch immer ich nach Deutschland zurück gereist bin.*

In this concluding paragraph, I'd like to thank some very important people in my life. I'd like to thank my parents Peter Schadd and Anneke Lely for their support and for helping me realize my own potential. I'd like to thank my brother Maarten for his helpful advice and my baby niece Mila for being adorable. Lastly, I'd like to express my gratitude for the support of Brigitte Schadd and Kurt Laetsch.

Frederik Schadd, 2015

Acknowledgments

The research has been carried out under the auspices of SIKS, the Dutch Research School for Information and Knowledge Systems. This research has been funded by the transnational University Limbug (tUL).

Table of Contents

Chapter 1

Introduction

1.1 Knowledge Systems and Information Exchange

Many technologies that emerged from the dawn of the information age rely on the accessibility of stored information. To be able to interpret the stored data, it is required that the data is structured and annotated with regard to its meaning and relation to other data entries. To formally specify how data in a knowledge system is structured, one typically creates an ontology. An ontology is defined *as an explicit specification of a conceptualization* (Gruber, 1993). In essence, given a domain, an ontology defines a list of concepts which exist in this domain, what data values are associated with each concept and how the concepts are related to each other. Depending on the technology used to express the ontology, an ontology description can even include logic-based axioms, facilitating logical reasoning over an ontology and the data that is encoded using that ontology.

The implicit intention behind the creation of an ontology is that it represents a consensus on how a given domain should be modelled with regard to its potential applications in a knowledge system. The term *consensus* implies that multiple parties are involved in the creation of the ontology. A party may contribute to the ontology development process by for instance suggesting terminology or relations, or by discussing or criticizing the suggestions of other parties. This process iterates until all parties are in agreement about the correctness and completeness of the ontology. After the consensus is reached, all involved parties can implement and populate their knowledge systems based on a single ontology. A globally shared ontology has the advantage that it facilitates the smooth access and exchange of information between different knowledge systems.

Ideally, ontologies should be shared in order to facilitate the interoperability between knowledge systems. Unfortunately, it is very common that businesses and developers choose to create their own ontology instead of adopting an already created ontology. This decision can be justified by multiple motivations. Technical motivations can be the particular knowledge system being required for different tasks, the developers following different design principles or the domain expert developing the ontology having a different perspective on the domain. Another motivation can be that a fully developed ontology represents a strategic asset to a company. Therefore, a company might be unwilling to share its ontology with a competitor. One option for preventing the development of many ontologies modelling the same domain involves the creation of a global standard, created by a consensus reached by domain experts, which can then be adopted by all relevant knowledge systems. However, it is possible that the experts tasked with the creation of a global standard do not agree on how the given domain should be modelled, leading to the situation where for a given domain a global standard does not exist. This leads to a common situation where the exchange of information between two knowledge systems is impeded due to the deployment of differing ontologies by the systems.

Instead or trying to create ontology standards, it is also possible to take a different approach by facilitating information exchange between knowledge systems utilizing different ontologies is the process of ontology mapping. This approach will be the main focus of this thesis. Ontology mapping involves the identification of correspondences between the definitions of two given ontologies. A *correspondence* denotes a certain semantic relation, e.g. equivalence or subsumption, which is ascertained to hold between two ontology concepts. It follows that a correspondence denotes that two given ontology concepts are likely used to encode the same type information. Given a correspondence between two concepts, one can develop a bi-directional function which is capable of transforming the encoded information of one concept such that it conforms to the specification of the other. The collection of all correspondences between two ontologies is referred to as an *alignment* or *mapping*. Thus, the identification of an alignment between two ontologies, which denotes all concept pairs which encode the same information, forms the foundation for the facilitation of information exchange between two heterogeneous knowledge systems.

Ontology mapping is applicable to various processes which regularly occur in businesses which maintain an information infrastructure or web-service based businesses. Example processes involving ontology-based knowledge systems which businesses need to perform are knowledge system mergers, ontology evolution, query answering or data translation. Since these processes aide businesses in their daily operations or realizations of long term goals, they are of strategical importance, therefore making ontology mapping a vital tool for the operation of a business.

An alignment between two ontologies is created by measuring the similarities between their modelled concepts. Highly similar concept-pairs are extracted which then form the alignment. To what extent two given ontology concepts are likely to denote the same information is measured by determining the overlap or similarities between their respective available meta-data. Exploitable meta-data can for instance be concept names or labels, structural information, or data that has already been modelled using that ontology. Specialized techniques examine some type of meta-data and quantify the similarity between two inputs. Many approaches have been developed for the purpose of determining concept similarities. While many approaches only utilize the information encoded within the ontology, some novel approaches attempt to utilize external information to aid in the comparison of ontology concepts.

This thesis aims at extending the available scope of approaches which utilize external information in order to compute concept correspondences. Specifically, the presented research focuses on two types of external information: (1) lexical resources and (2) partial alignments. The first type, being lexical resources, can be described as an enriched thesaurus. A thesauri typically consists of a list of word definitions. Each word is commonly defined by a set of synonyms and a written explanation. Next to the word definitions, a lexical resource also contains relations which hold between the different words, allowing a reader to quickly query related definitions in order to gain more information about a certain word. For example, a very popular thesaurus is *Wikipedia*, with the English variant containing more than 4 million articles. The second category of investigated external information is a special type of alignment. An alignment between two ontologies is a complete list of all concept-pairs which can be used to model the same information. A *partial alignment* between two ontologies however is an incomplete list of concept-pairs. These can be the result of a domain expert attempting to create an alignment but being unable to finish it. Therefore, partial alignments can be exploited in order to create complete alignments.

1.2 Applications

Comparing and mapping meta-data of structural models is a common task occurring in a variety of fields. This task has its origins in the field of databases, where the objective is to match database schemas such that one can transfer data between two databases (Batini, Lenzerini, and Navathe, 1986; Kim and Seo, 1991). In the past decade, newer technologies for the representation and application of ontologies became available. With their introduction, the problem of enabling information exchange became more prevalent. We will provide an overview of a selection of notable applications where ontology mapping can provide a viable long-term solution for potential ontology interoperability problems. In order, the discussed applications are (1) schema integration, (2) information integration, (3) ontology engineering, (4) information sharing, (5) web-service composition, (6) Natural-language-based querying of semantic information and (7) agent communication.

1.2.1 Schema Integration

Schema integration (Batini *et al.*, 1986; Sheth and Larson, 1990; Sheth, Gala, and Navathe, 1993; Batini and Lenzerini, 1984; Spaccapietra and Parent, 1994; Parent and Spaccapietra, 1998) is the oldest process which involves the task of ontology mapping. In this category, the general problem is that two parties, each possessing a knowledge resource with corresponding ontology, wish to establish the exchange of information between their respective resources. This exchange of information can be unidirectional or bidirectional.

Figure 1.1 depicts a generalized schema integration task. Here, *Party 1* and *Party 2* each operate a knowledge resource in which information is stored, possibly using

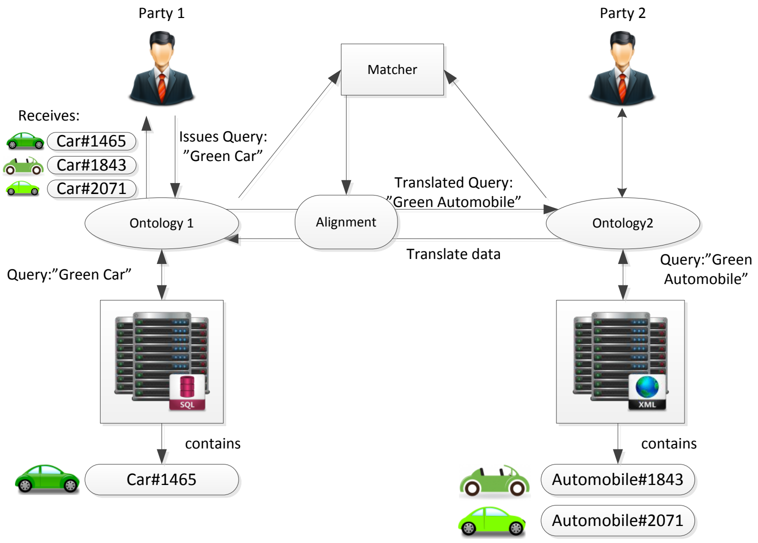


Figure 1.1: Example illustration of a schema integration task.

different technologies such as SQL[1], XML[2], XSD[3], RDF[4] or OWL[5]. Each party has encoded its information using an ontology which fulfils the party's own needs, in this example represented as *Ontology 1* and *Ontology 2*. The general task of information integration involves the creation of an alignment between the two ontologies, such that each party can gain access to the data stored in the other party's information system. This is illustrated by *Party 1* issuing a query concerning a *Green Car*. The created mapping allows for the translation of the query into the terminology of *Ontology 2*, such that the example query is now reformulated as *Green Automobile*. The translated query can then be executed on the ontology belonging to *Party 2*, resulting in a list of relevant data entries. These entries however are still expressed

[1] Structured-queryl-language(SQL). Query language for accessing and managing relational databases. Commercially released in 1979 by Relational Software Inc. (now Oracle Corporation), it became widely used in the following decades as the standard data storage and management technology.

[2] Extensible Markup Language (XML). Markup language for storing information in a structured and machine-readable way. Since it is a text-based storage format, it is widely used to transfer structured data over the internet.

[3] XML-Schema-Definition(XSD). Schema language for defining how specific types of XML documents should be structured.

[4] Resource-Description-Framework(RDF). Data-model based on making statements about resources. Statements are formulated as *triplets* in a subject-predicate-object format, with a collection of triples essentially forming a directed and labelled multigraph. Data in an RDF storage can link to resources of other storages.

[5] Web-Ontology-Language(OWL). Language for specifying ontologies based on formal semantics. A variety of syntaxes exist for expressing OWL ontologies, for instance utilizing XML or RDF.

using the terminology of *Ontology 2*, meaning that these cannot yet be processed by the knowledge system of *Party 1*. However, using the alignment between *Ontology 1* and *Ontology 2*, all retrieved data entries can be translated back into the terminology of *Ontology 1*, such that these can be presented as query results to *Party 1*.

The process of schema integration is performed regularly in the corporate field. A company's knowledge system is typically the responsibility of the company's Chief Information Officer (CIO). One of these responsibilities is Enterprise Meta-data Management (EMM), which includes the design and maintenance of the ontology of the company's knowledge system. Should a company wish to gain access to an external knowledge system, it becomes the responsibility of the CIO to oversee the integration of the new knowledge system.

Suppose that two companies wish to merge their operations, or that one company performs a takeover of another company. Both companies are likely to possess knowledge systems storing critical information, like customer information or product data. Here, the critical task is that the two knowledge resources have to be merged into a single resource. In a takeover scenario this is typically the resource belonging to the company performing the takeover. As the first step, a mapping needs to be created between the two ontologies. These are likely to differ as the two companies had different requirements for their own resources during their creation. These differences can stem from the system designer following different design principles, the companies operating in different areas and even from the companies having different business goals, which might have to be adjusted after the merger/acquisition.

Alternatively, two companies may decide to strategically cooperate, requiring the free exchange of information between their respective knowledge resources. In such a scenario, each company needs access to the other company's respective information without having to alter its own internal infrastructure. A mapping would thus allow a company to seamlessly access the partner's company data.

1.2.2 Information Integration

Information integration is another common task which requires a mapping between ontologies. Processes which we define under information integration are catalogue integration (Bouquet, Serafini, and Zanobini, 2003; Ding *et al.*, 2002), data integration (Halevy, Rajaraman, and Ordille, 2006; Arens, Knoblock, and Shen, 1996; Calvanese *et al.*, 1998; Subrahmanian *et al.*, 1995; Halevy *et al.*, 2005; Lenzerini, 2002) and data warehousing (Bernstein and Rahm, 2000). Here, access to heterogeneous sources is facilitated by the creation of a common ontology, also referred to as a mediator. In the following paragraphs we will provide a general illustration of an information integration scenario. In subsections 1.2.2 and 1.2.2 we will discuss catalogue integration and data integration in more detail.

Figure 1.2 illustrates an example scenario of an information integration problem. One is given a series of knowledge resources, with every resource being encoded using its own specific ontology, denoted here by *Local Ontology 1* and *Local Ontology 2*. The goal is to provide the user with a single interface with which a multitude of resources can be queried. The accessible resources can also be implemented using different technologies such as SQL, XML, RDF or OWL. A *Common Ontology* is

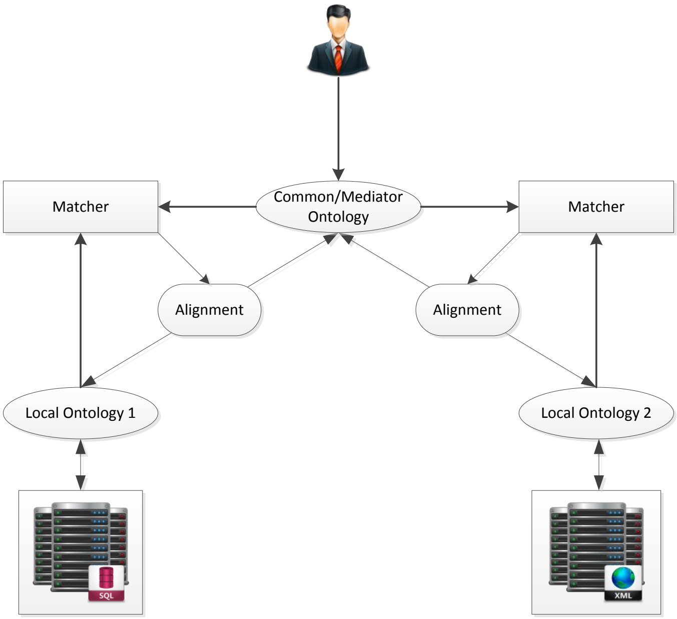


Figure 1.2: Example illustration of an information integration task.

created which models all information that is encoded using the local ontologies. Creating a mapping between the *Common Ontology* and all local ontologies would allow the user to formulate a query by only using the terms of the *Common Ontology*. That query can then be translated into the terminologies of each local ontology, thus enabling access to all local ontologies. After the query is executed on each individual resource, all retrieved search results are translated back into terms of the common ontology and merged before being presented to the user. Depending on the domain, the process of merging these multiple result lists might remove duplicate results or specifically contrast very similar results so they can be compared easier.

For example, a user might desire the answer to the query *"find a DVD with the title 'Sherlock' "*. When entered into an information system modelling media, the query is then translated into the ontology-terminology used by the different connected resources, for example *amazon.com* or *bol.com*. The information integration system would execute the translated queries on each resource, translate the results back into the terms of the common ontology and present a merger of all results to the user. Note that at no point was it necessary for the user to directly interact with the individual resources, interaction was limited to the interface provided by the integration system.

The problem presented in sub-section 1.2.1, denoted as *schema integration*, is very related to the task of information integration, despite the fact that it typically does not involve the creation of a common ontology. If one were to interpret one of the given ontologies as a common ontology, then the entire system can be interpreted as an information integration system where one local ontology is already interoperable with the common ontology. Also, it may be the case that the two ontologies do not overlap perfectly with regard to the scope of the modelled data. For example, given two ontologies A and B modelling the pharmaceutical domain it may be the case that herbal medicine is modelled in A and experimental medicine is only modelled in B. In this scenario, designating either A or B as mediator ontology would require an update to the mediator such that the information of the remaining ontology is modelled as well. This updated version of A or B can thus be interpreted as the mediator ontology of an information integration system.

Information integration systems share some similarities with *Federated Search* systems. A *Federated Search* system is a type of meta information-retrieval-system, where user queries are distributed among local collections (Shokouhi and Si, 2011). A local collection represents a single search engine, such as *Google* or *Yahoo*. The results of the local collections are merged into a single list and presented to the user. There are however some key differences between *Federated Search* systems and information integration systems. First, both local collections and federated search systems can only process queries that are expresses as simple strings. These strings typically contain keywords or a natural language phrase. Therefore, any query is compatible with any search system, since the query does not need to be translated into a different terminology. The quality of the search results thus depends on the strength of the algorithms deployed by each local collection. While a *federated search system* does not employ a global ontology, it does utilize a management system referred to as the *Broker*. The *Broker* is responsible for determining which local collections are relevant to a particular query, forwarding the query to each relevant local collection, retrieving the search results of each local collection and merging different result lists into a single list. The second key difference is that the retrieved results are a series of documents rated by relevance instead of a list of data entries. Thus, if a particular piece of data is required, the user must still inspect every document until the desired data is found.

Catalogue Integration

Businesses which produce and sell a vast array of different products are likely to store information about their products in a catalogue. Here, the ontology used to model the product data is designed for the specific goal of organizing and classifying their products. An ontology that is designed for such a goal is also referred to as hierarchical classification (HC). In a business-to-business application (B2B), a company selling products from its catalogue might wish to sell its products on a marketplace. Marketplaces, such as *eBay* or the *Amazon Marketplace*, have current offerings organized using their own HC. Thus, if a seller wishes to automatically sell his products on the different available marketplaces, it becomes necessary to generate a mapping between the seller's HC and the HC of each marketplace. This task is referred to as the catalogue matching problem (Bouquet *et al.*, 2003; Ding *et al.*, 2002). With such mappings in place, a seller is able to automatically translate product descriptions and specifications into the terminology of the marketplace HC and submit products to that marketplace for sale to customers.

There have been initiatives to standardize product classifications by creating a global HC. A global HC would then model all types of products such that if one is able to express product details using the global HC terminology, one would then be able to transfer information between all catalogues which support the global HC. Examples of such global HC's are *UNSPSC*, *eCl@ss*, *eOTD* and the *RosettaNet Technical Dictionary*. Related standards can be found in trade-item-identifier systems, such as the *International Article Number* (EAN) system and the *Universal Product Code* (UPC) system. These systems provide a unique numerical value for every individual product . Products are fitted with a depiction of their respective item code in the form of a bar-code, such that the trade of these products can be easily tracked in stores. Contrary to a HC, a trade-item-identifier system does not contain a hierarchical structure with which products can be classified and organized. If a knowledge system of a company however does not support the global HC which is adopted by a certain marketplace, then that company would have to create a mapping between its own catalogue and the global HC such that the company can translate its product descriptions such that it can be entered in the marketplace.

Even if a knowledge system has adopted a specific global HC, it might differ from the HC deployed in the marketplace. This can stem from the marketplace having adopted a different HC or having developed its own. An additional issue is that the creation and maintenance of such global HCs presents challenges. Often they are unevenly balanced, lack in specificity and require a significant amount of maintenance in order for the HCs to keep up with current developments (Hepp, Leukel, and Schmitz, 2007). Submitting suggestions, requests or additions to a central governing entity that manages a global HC can be a time consuming process, which can cause frustration for companies having adopted a HC and wishing for changes that suit their personal needs. Lowering the barrier for submitting changes could potentially alleviate this problem. The *OntoWiki* (Hepp, Bachlechner, and Siorpaes, 2006) project demonstrates that a de-centralized system, here comprised of several Wikipedia communities, is a possibility for the creation of a consensual global categorization.

Given that a global HC is constantly being managed and updated, the adaptation of a global HC would also mean that eventually an update might induce changes as a result of which the current terminology is no longer compatible with the updated version. Here, a mapping between the two HCs is required in order to restore compatibility. This type of problem is further discussed in subsection 1.2.3.

Data Integration

Data Integration (Halevy *et al.*, 2006; Arens *et al.*, 1996; Calvanese *et al.*, 1998; Subrahmanian *et al.*, 1995) is a special type of information integration problem, of which the key aspect is that the data is not fully loaded into a central information system before the exchange of information (Halevy *et al.*, 2005). Here, a mapping for each element of a source ontology is denoted as a query over the target ontology. One can distinguish these mappings as a Local-as-View (LAV) and Global-as-View

(GAV) approach (Lenzerini, 2002).

In a GAV approach, a mapping between the *global ontology* and a *local ontology* is formulated such that each concept of the *global ontology* is mapped with a query over the *local ontology.* As an example, assume that a global ontology G models the concept *vehicle* and that a local ontology L models the concepts *car* and *motorcycle.* In a GAV approach a mapping for the concept *vehicle* can be expressed as:

$$\text{vehicle} \leftrightarrow \textbf{SELECT}\ ^{*}\ \textbf{FROM}\ \text{car},\ \text{motorcycle}$$

This principle tends to be more advantageous if the local ontologies are considered stable, i.e. experience little to no changes or updates. The LAV approach differs to GAV with regard to which ontology concepts in a mapping are denoted as a query. Here, the concepts of the *local ontologies* are expressed as queries over the *global ontology* in the mapping. Returning to the previous example, a LAV mapping for the concept *motorcycle* from the local ontology L can be expressed as:

$$\textbf{SELECT}\ ^{*}\ \textbf{FROM}\ \text{vehicle}\ \textbf{WHERE}\ \text{wheels} = 2 \leftrightarrow \text{motorcycle}$$

This approach is effective if the global ontology in the information system is stable and well established, preferably as a global standard for the given domain.

In both the LAV and GAV approach, the queries are processed using an inference engine, which allows the query to be expressed using the terminology of each local ontology. In a Data Integration system the data of each local ontology is not fully loaded into a central information system. This has the distinct advantage that the relevant data of each query is retrieved during the execution of each query. For a user this means that he has always access to the most up-to-date information across the entire system.

1.2.3 Ontology Engineering

Ontology Engineering is the process of building and maintaining an ontology for a desired knowledge system. Particularly the maintenance aspect of ontology engineering might require the application of mapping techniques. For knowledge systems that are deployed over a long period of time, it can occur that certain changes have to be made. These can be small and incremental for the purpose of improving the ontology, causing it to evolve over time. More abrupt and significant changes might also occur, especially when there is a change in the requirements or a shift of corporate strategy (Plessers and De Troyer, 2005; Rogozan and Paquette, 2005; Redmond *et al.*, 2008). Furthermore, if the ontology is distributed among multiple users, it can occur that each user develops different requirements over time. Each user would then wish to update the ontology such that it suits his needs. This can lead to different versions of the same original ontology being deployed among multiple internal or external information systems. Often the individual changes serve the same purpose but are executed differently, due to the system designers of a single information system not having access to the change-log of other information systems. This causes the designers to be unaware of any changes that have already been performed across the different systems. Hence, it can occur that sufficient changes to an ontology cause it to be incompatible with its original version (Klein and Noy, 2003; Noy and Musen, 2002; Hepp and Roman, 2007; Noy and Musen, 2004). In this case one can perform ontology mapping so that one can transfer the data encoded using the original version into a system using the updated ontology.

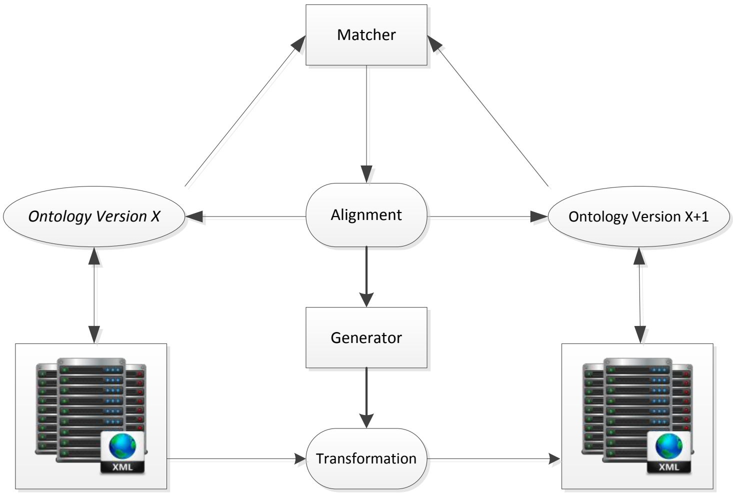


Figure 1.3: Example of an ontology engineering task.

A general mapping task in an ontology engineering scenario is presented in Figure 1.3. An XML based knowledge system has its data encoded using *Ontology Version X*. At some point in time changes are made to that ontology, resulting in the creation of *Ontology Version X+1*. This ontology encodes the information of the same domain, but in a slightly different way. For instance, in the new ontology entities could have been added, removed or renamed or data values could have been altered such that these use different data-types or model data up to a different accuracy. In this scenario, an ontology mapping approach needs to find all corresponding concepts between the old and new ontology. Based on the generated mapping, a *transformation* needs to be created which dictates how data needs to be processed such that it conforms to the new ontology. Using the transformation, all data instances can then be converted such that these conform to the new ontology.

1.2.4 Information Sharing

A different type of information exchange infrastructures are Peer-To-Peer (P2P) systems (Androutsellis-Theotokis and Spinellis, 2004; Ehrig *et al.*, 2004; Pouwelse *et al.*, 2005). In a P2P system content is spread across different nodes. These nodes can be internal servers or spread around the globe. The nature of P2P systems provides several distinct advantages over centralized systems, such as easy scalability and robustness against attacks or defects, resulting in the benefit that information stays available even if nodes are taken off-line for a given reason. Thus, a distinct feature of a P2P system is that it is decentralized. One can categorize the degree of decentralization in three categories (Androutsellis-Theotokis and Spinellis, 2004):

Hybrid Decentralized A set of client systems, with each system storing information that is available over the entire network. The clients are connected to a central directory server which maintains a list of clients and a list of files that each client offers. A given client desiring information connects first to the central server to obtain a list of connected clients. Then, the given client individually connects with each client in the network in order to obtain the required information.

Purely Decentralized Each node in the network acts both as a client and a server, meaning that there is no central dedicated server which coordinates network activity. Network activity is propagated using a broadcast-like mechanism, where a node upon receiving a message, for example a file request, will not only answer the message but also repeat it to its neighbours. The neighbour responses are then back-propagated through the message trace until they are received by the original sender.

Partially Centralized A partially centralized system is similar to a purely decentralized system in that it has no central dedicated server. Instead, nodes are dynamically assigned the task of *super-node* based on the bandwidth and computing power of the node. A super-node acts as a server for a small cluster of nodes, indexing its files and possibly cashing the files of its assigned nodes. By acting as a proxy, the super-node initially handles many of the search requests on behalf of its assigned nodes, thus reducing the computational load of these nodes.

Some P2P systems, such as *BitTorrent*, *eDonkey* or *Gnutella*, describe their content using a globally adopted ontology, where files are for instance annotated using attributes such as *'name'*, *'title'*, *'release date'* or *'author'*. Thus, the problems one encounters when faced with heterogeneous ontologies are circumvented by enforcing an ontology over all connected information sources. However, for a P2P-based data exchange on the semantic web it is likely the case that the enforcement of a global ontology is a undesirable solution due to the autonomy of each node being of importance (Nejdl *et al.*, 2002; Ives *et al.*, 2004). Here, instead of individual people the nodes of the P2P system would represent the knowledge systems of companies or organizations, who are likely to have different requirements for their own system and might be unwilling to convert their system to a different ontology for the sole purpose of joining a P2P network. If the nodes of a P2P network utilize difference ontologies, then it is necessary that the nodes are able to map their respective local ontologies in order for queries and information to be successfully transferred across a P2P network. Such a scenario is illustrated in Figure 1.4 for a *hybrid decentralized* system.

In this scenario, a client using *Local Ontology 1* would connect to a central directory server. This server returns a list of known clients. The clients establish

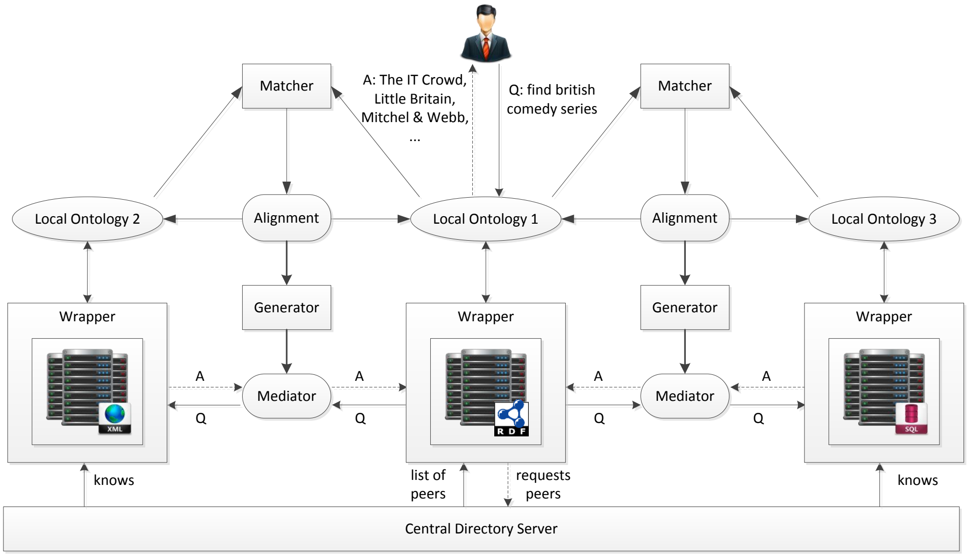


Figure 1.4: Information sharing in a hybrid decentralized P2P system.

communication with each other and the server using a specific P2P protocol, which is implemented as a wrapper for the knowledge system of each client. In the case that the information of the retrieved clients are encoded using a different ontology, in this example *Local Ontology 2* and *Local Ontology 3*, the client would need to map its ontology with the ontologies of all other clients within the network. Once the mappings are created, the given client can then send appropriately translated queries to all clients in the network and interpret their responses. Note that without a mapping process as described subsection 1.2.1 it is only possible to exchange information with clients using the same ontology, meaning that only part of all information residing in the network is available to the user.

For *purely decentralized* networks the ability to communicate between clients is also a very vital issue. Here, any given client has only a limited amount of direct connections to other clients within the network. The remaining clients within the network are reached by forwarding and back-propagating queries and answers. If the directly connected clients do not utilize the same ontology, then this means that the given client cannot submit appropriately formatted queries, resulting in the client essentially being completely isolated from the rest of the network. The analytical metrics to measure properties of networks, including vulnerabilities and reachability of nodes, originate from the field *network science* (Börner, Sanyal, and Vespignani, 2007). For example, measures of *centrality* can be used to express the reachability of one or multiple nodes within the graph, whereas the measure of *modularity* expresses how strongly a network is separated into different groups. For network with a high modularity it is of high importance that nodes connecting different clusters can exchange information. Otherwise entire clusters of nodes can be cut off from the rest of the network.

Significant issues can also occur in a *partially centralized* network. If a node utilizes a different ontology than its assigned *super-node*, then this would effectively result in the entire network being unable to access the information of that node. This occurs due to the *super-node* being unable to index the information of the given node. The problem of inaccessible information is exacerbated if the ontology of a super-node differs from the ontologies of the other super-nodes in the network. Such a situation would result in an entire cluster of clients being isolated from the system, which could eventually completely fragment the network if a sufficient quantity of different ontologies are employed by the super-nodes.

1.2.5 Web-Service Composition

The original capabilities of the internet were designed as a tool for the global access of information by humans. With the emergence of *Semantic Web* technologies it also became possible for businesses to offer services directly via the internet (Bussler, Fensel, and Maedche, 2002). The *Semantic Web* envisions web-services expressing their capabilities using precise semantics, such that autonomous agents that roam the web can automatically discover these services, interpret their functionality, decide which service best suits the agents' needs and interact with the most appropriate service. The availability of web-services means that businesses can effectively outsource certain components of their applications, for example querying services or data processing, to web-services which specialize in specific areas and tasks. Thus, a service offered by a business can essentially become a composite of interconnected services.

The process of Web-Service Composition can be described by three core tasks which need to be performed (Sycara *et al.*, 2003): (1) a *planning* task, where all processes in the application are specified, especially with regard to the inputs and outputs of certain components, (2) a *discovery* task, where web-services are discovered which can execute the tasks described in the components of the *plan*, and (3) the management of *interaction* between the identified services. Of those three tasks, two of these might require the application of mapping techniques in order to resolve certain heterogeneities. The most obvious application is the task of web-service *integration.* Once a service is identified, it may become necessary to perform a mapping step in order to be able to translate queries into the input format specified by the ontology which is used by the accessed web-service. This task is comparable to the process of *information integration*, described in section 1.2.2.

The task of web-service *discovery* requires more unique mapping techniques (Medjahed, Bouguettaya, and Elmagarmid, 2003; Maximilien and Singh, 2004; Klusch, Fries, and Sycara, 2009). In order for a web-service to be discovered by an agent, a semantic description needs to be matched with a designed profile of the desired service. Service descriptions are typically expressed using specialized ontologies, such as DAML-S[6] (Coalition *et al.*, 2002) and OWL-S[7] (Martin *et al.*, 2004). The first issue that arises here is that the ontologies used to describe the service capabilities need to be interoperable, such that all details of a service are expressed using the same terminology. It can occur that a web service expresses its capabilities using a different ontology than the one used by the business performing the service composition. Thus, in this case *schema integration* needs to be performed between the two service description ontologies.

Mapping the service description ontologies ensures a compatibility with regard to how a service description is expressed. As a next step, all encountered descriptions need to be translated and mapped to the required service description so that the most appropriate service can be found. Here, all instances of the service ontology and their associated attribute data, referred to as a service profile, need to be examined and compared to the profile representing the desired service (Medjahed *et al.*, 2003; Klusch, Fries, and Sycara, 2006). This problem is essentially a soft version of an *instance mapping* problem (Ferrara *et al.*, 2008). In *instance mapping*, one is given two instances belonging to the same ontology or two different ontologies and one has to determine whether or not the two given instances denote the same object. While in instance mapping the desired output is a *'true'* or *'false'* statement, in the case of mapping service profiles one would need to express the output as a continuous value, since one would rather choose a service being capable of almost all required tasks than a service capable of none of them, despite neither being able to perform all desired tasks. The processes of instance mapping and ontology mapping are very interrelated, since one can utilize instance mapping to solve an ontology mapping problem and vice-versa (Wang, Englebienne, and Schlobach, 2008). Additionally, both problems have theoretical similarities, where in both cases the inputs are two lists of entities, lists of concepts for ontology mapping and lists of individual entities for instance mapping, and the mapping system needs to exploit the available metadata in order to determine the overlapping entities between the two lists. A result of this interrelatedness is that systems which tackle either ontology mapping or instance mapping problems often exhibit a significant overlap with regards to their applied techniques.

Figure 1.5 depicts a web-service composition scenario. A given *web-application* wishes to outsource a component of its system to a web-service. To achieve this, it creates a *service description* which describes the ideal properties of the desired system. In this example, the description is formulated using the DAML-S service ontology. Any web-service on the internet might advertise its capabilities using a different service ontology, however. In this example, the ideal *web-service* has expressed its capabilities using OWL-S. Therefore, in order to be able to determine whether the service is appropriate, the terminology of OWL-S must be matched to DAML-S such that the service descriptions can be compared. After a mapping between the two service ontologies is established, a *translator* is generated which is

[6] DARPA Agent Markup Language for Services (DAML-S). Built using the DAML+OIL ontology language. DAML+OIL combines features from both the DARPA Agent Markup Language (DAML) and the Ontology Inference Layer (OIL).

[7] Built using the Web-Ontology-Language(OWL). OWL is used for specifying ontologies based on formal semantics.

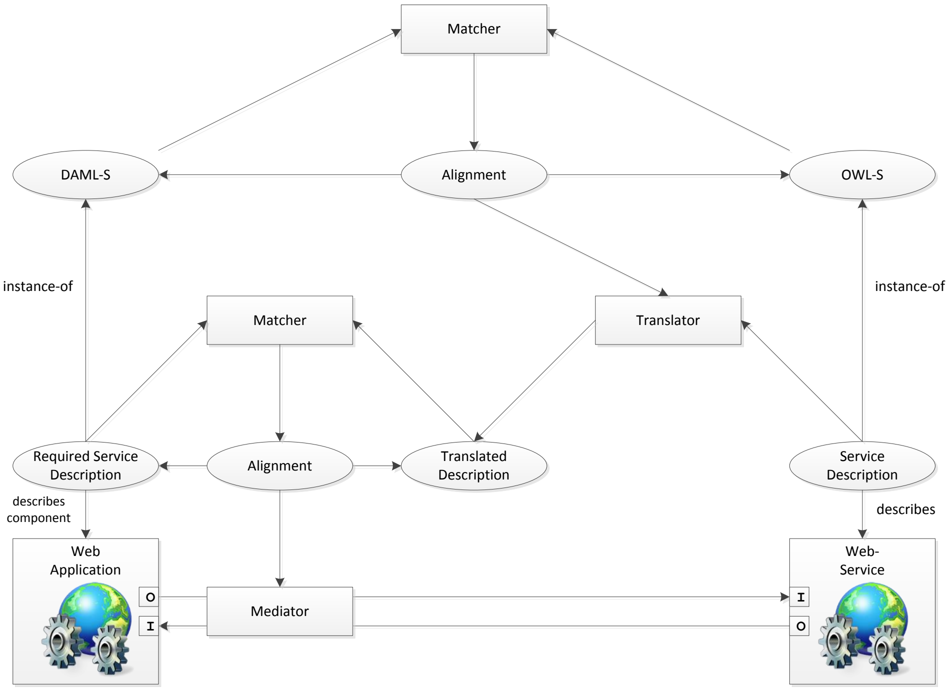


Figure 1.5: Mapping tasks in a web-service composition scenario.

capable of translating service descriptions into the different ontology terminologies. This translator then reformulates the service description into the DAML-S format. Using the *translated description* a matching system specialized in comparing service descriptions compares the translated description with the description of the desired service. This comparison entails an estimate as to how appropriate the service is for the desired task and how the different inputs and output should be mapped such that the application can interact with the service. Once the ideal web-service has been determined, a mediator is created which, based on the mapping between the service descriptions, can translate and transfer the inputs and outputs between the application and the web-service.

1.2.6 Querying of Semantic Information with Natural Language

In the subsection regarding *information integration* we described a scenario where a user would formulate a query using the terminology and semantic properties of a common ontology. In essence, the effectiveness of such a system relies on the familiarity of the user with the applied common ontology with which queries are formulated. For specialized applications intended for businesses a familiarity of the user with query-formulation can be assumed. However, one cannot assume that the user is comparably familiar with queries when the application is intended for the general public. For a user of the general public one can only assume a familiarity with *Information Retrieval* (IR) systems, services such as *Google*, *Bing* or *DuckDuckGo*, which only receive *natural language* (NL) or keywords as queries. For such a user-base to be able to effectively query semantic information sources one must parse the natural language query into an ontology-based format, similar to queries executed in information integration systems.

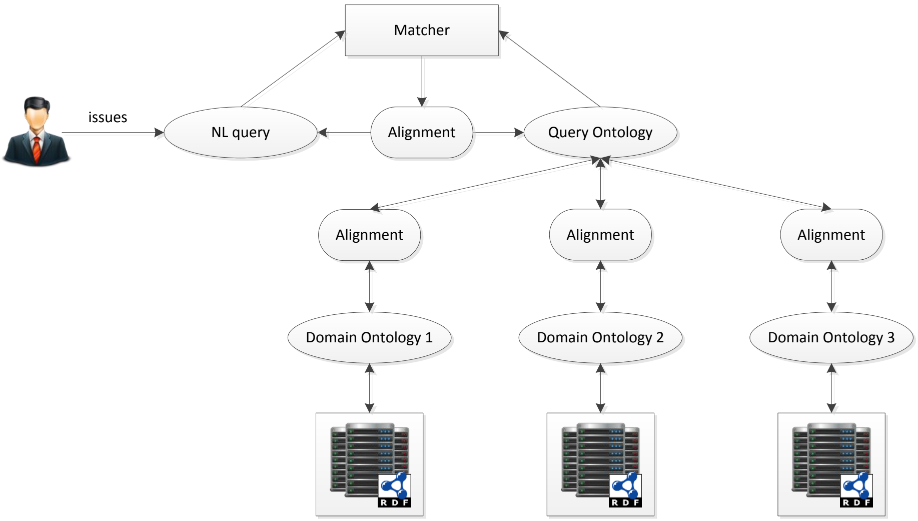


Figure 1.6: Mapping in an information system receiving NL queries.

The process of parsing a natural language query into an ontology-based query can be interpreted as a mapping task. However, one of the ontologies is limited to the terminology occurring in the query and the only available structural information is the word order in which the query was written. Thus, in essence this is a mapping task in which one ontology contains significantly less meta-information for each concept than professionally engineered ontologies. To complement the techniques used in standard *information integration* scenarios, special approaches are developed which individually process each word, e.g. by grammatical and semantic annotation, and create a mapping between the user input and a query ontology. From this mapping a semantic query is created using the terminology of the query ontology (Lopez, Pasin, and Motta, 2005; Tran *et al.*, 2007; Lei, Uren, and Motta, 2006). A decision system then forwards the generated query to accessible knowledge resource which might contain information relevant to the query (Lopez *et al.*, 2007). Executing a query formulated using the terminology of a *query ontology* over different system is an example of a *information integration* problem, where one can utilize the techniques which are often applied in this situation in order to access the available knowledge resources.

A noteworthy example of a natural language query system is *Watson*, developed by IBM Research (Ferrucci *et al.*, 2010). Watson is a special-purpose knowledge system designed for the *Jeopardy!* challenge, intended to be equally ambitious as IBM's previously conquered challenge of defeating a world-champion chess player using the *Deep Blue* supercomputer (Hsu, 2002). The goal of the *Jeopardy!* challenge was to design a question and answering (QA) system which could outperform human champions on the TV game-show *Jeopardy!*. *Jeopardy!* is essentially a trivia-based question and answering game-show with certain unique gameplay properties. The most recognizable feature is that instead of questions the contestants receive a piece of trivia about an unknown entity. The contestants must guess which entity is being referred to in the trivia and must respond with the entity, while formulating his response as a question. For example, a contestant may receive the following trivia: *'In 1996, he wrote the novel 'A Game of Thrones', which was later adapted into a TV series by HBO.'*. Based on this trivia, the contestant must guess that the intended entity is the author George R. R. Martin and phrase his answer as *'Who is George R. R. Martin?'*. The game is structured around a board of categories, with each category containing a series of options ordered in increasing monetary worth. The categories can be broadly defined, e.g. *'History'* or *'Literature'*, or only cover specific topics, e.g. *'Japan US Relations'* or *'Potent Potables'*. Contestants can choose options from categories that they are familiar with, however the specific categories are unknown before the show. A player can earn the monetary value of a question by answering it correctly. However, attempting to answer any question is always associated with a risk. If a contestant fails to answer a question correctly then the monetary value of that particular question is subtracted from the contestant's earnings. It is thus important that a contestant is certain of his answers.

The difficulty for a query system in this domain is that it needs to produce a single result with a high degree of certainty. This is significantly more difficult than the task of information retrieval, where a set of results of which most individuals are 'relevant' to some degree can be seen as a good output. The Watson system parses an input query into a linguistic model which describes the grammatical roles of words and their relations to other words in the query (McCord, Murdock, and Boguraev, 2012). Relational-queries which could denote the intended query are created using the information of the parsed user query. When given a specific piece of trivia, a linguistic analysis system first identifies the grammatical type of each word and parses the sentence into a linguistic model. Based on this analysis, it is identified which domains are relevant to the query and candidate semantic relations are gathered which form the basis for potential queries. Examples of such relations are *author-of*, *appeared-in* and *produced-by*. The *author-of* relation can be structured like *author-of::[Author][verb][Work]*, meaning that if the mapped terms of the input query match with some of the classes of the relational query, then the missing term could denote the answer to the query. A ranking system ranks all extracted relational-queries according to their likelihood of denoting the intended query through relational analysis (Wang *et al.*, 2012).

1.2.7 Agent Communication

Agents are autonomous software-driven entities designed to independently perform tasks and solve problems. The domains in which agents are deployed are very divergent, with approaches being developed for negotiation (Chen and Weiss, 2012), power system restoration (Nagata *et al.*, 2000), resource allocation (Chavez, Moukas, and Maes, 1997), organ-transplant coordination (Aldea *et al.*, 2001), e-commerce (Xiao and Benbasat, 2007), cloud computing (Cao, Li, and Xia, 2009) or smart grids (Pipattanasomporn, Feroze, and Rahman, 2009). Agents communicate with each other using special communication frameworks, for example KQML[8] (Finin *et al.*, 1994) or FIPA-ACL[9] (Labrou, Finin, and Peng, 1999; FIPA, 2008), which allows messages to be annotated with an interaction-specific context, for instance *'agree'*, *'disagree'* or *'request-information'*. The actual content of messages is likely expressed using knowledge-representational languages and using the terminology of a specific domain ontology. Thus, if two agents interact by exchanging messages using different terminologies, then there is only a very small chance that these agents will be able to achieve any meaningful interaction or reach an agreement.

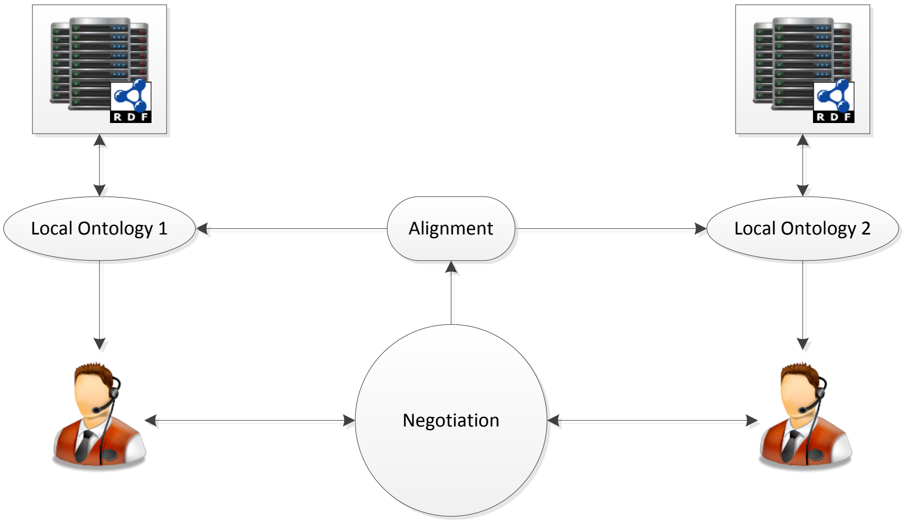


Figure 1.7: Mapping in an agent communication scenario.

In the case that two interacting agents have a different ontology in which they express their information, they must first map their ontologies in order to achieve a meaningful interaction. For this, they must autonomously communicate their ter-

[8] Knowledge Query Manipulation Language (KQML).

[9] Agent Communication Language (ACL) developed by the Foundation for Intelligent Physical Agents (FIPA).

minologies and reach a consensus on how each term should be mapped. Typical approaches here revolve around argumentation techniques, in which agents argue in what ways mappings can be established or conflict with other mappings (Trojahn *et al.*, 2011), or to what extent their respective data overlaps (Wiesman, Roos, and Vogt, 2002; Wang and Gasser, 2002; Wiesman and Roos, 2004). The process of two agents establishing an alignment between their respective ontologies is illustrated in Figure 1.7.

1.3 Core challenges within Ontology Mapping

Evidenced by the previous section, ontology mapping can be applied as a solution to interoperability in various scenarios. An obvious challenge for all these scenarios is the quality of the produced alignments. Incorrect correspondences can either cause the data exchange between two systems to be erroneous or increase the overhead caused by verifying the produced alignments. Over past decade, certain specific aspects of the process of ontology mapping have emerged which have accumulated considerable research interest (Shvaiko and Euzenat, 2008; Shvaiko and Euzenat, 2013). We can categorize these aspects into seven challenges, namely (1) efficiency, (2) mapping with background knowledge, (3) automatic configuration, (4) user involvement, (5) correspondence justification, (6) crowdsourcing and (7) alignment infrastructures. Some of the listed challenges focus on distinct techniques with which alignment quality can be improved, e.g. mapping with background knowledge and automatic configuration, while other categories of research aim at improving the ontology mapping process in a non-qualitative way, e.g. efficiency or user involvement. We will provide an overview of each challenge in the following subsections.

1.3.1 Efficiency

Next to the quality of the produced alignments, the computational efficiency of the mapping systems is also of importance in many applications. Examples of such applications are mapping problems where the response-time is fixed in the given domain and the system must produce a mapping within a given time-frame. For instance, a human issuing a query to a QA system is unlikely to be willing to wait a long amount of time for a response. It is therefore of importance that ontology mapping solutions are computationally efficient such that these can be seamlessly integrated into a knowledge application. Some mapping systems are able to solve runtime issues through the copious use of memory, however it has been shown that this design choice can lead to memory bottlenecks (Giunchiglia, Yatskevich, and Shvaiko, 2007). Hence, memory consumption should be taken into account when developing approaches aimed at improving the runtime. A related application is the mapping of large-scale ontologies. A large-scale ontology can be defined as an ontology consisting of at least 1.000 concepts, though in certain domains ontologies can reach a size of 100.000 concepts. Computational efficiency is imperative here since, due to the large problem space. Hence, applying inefficient methods can easily result in a significantly increased computation time.

The necessity of computational efficiency has been recognized by the research community. The Ontology Alignment Evaluation Initiative (OAEI), which hosts yearly competition for the evaluation of mapping systems, added test sets consisting of large-scale mapping tasks, namely *Anatomy* and *Large BioMed*, to specifically evaluate the efficiency of mapping systems (Grau *et al.*, 2013; Euzenat *et al.*, 2011b; Euzenat *et al.*, 2010). Some research groups have responded to the challenge by developing light-weight versions of their existing systems, as seen with *LogMapLite* (Jiménez-Ruiz, Cuenca Grau, and Horrocks, 2012a) and *AgreementMakerLight* (Faria *et al.*, 2013) systems. Some systems, such as *QOM*[10] (Ehrig and Staab, 2004) tackle the efficiency problem by applying very efficient mapping strategies, while systems such as *GOMMA*[11] (Gross *et al.*, 2012) also exploit the scalability of available computational resources.

1.3.2 Mapping with background knowledge

An ontology is typically designed with specific background knowledge and context in mind. However, this type of information is rarely included in the ontology specification, which can cause difficulties in the mapping process. To overcome this issue, the main challenges are to discover and exploit missing background knowledge. The most prolific areas of mapping with background knowledge include the following areas:

Axiom enhancement Declaring missing axioms manually (Do and Rahm, 2002) or exploiting the axioms of available partial alignments (Lambrix and Liu, 2009).

Alignment re-use Exploiting the alignments of previous mapping efforts of the given ontologies (Aumueller *et al.*, 2005). Storing and sharing alignments facilitates the possibility of composing alignments using several pre-existing alignments. Given alignments linking both given ontologies to a third ontology one can derive an alignment through logical inference.

Internet-based background knowledge Exploiting internet-based resources and services to aide the mapping process. Specifically, one can utilize web-based linked-data structures (Jain *et al.*, 2010) or internet search engines (Gligorov *et al.*, 2007). For example, search engines can be utilized by analysing the probability of two concept names co-occurring in the search results.

Lexical background knowledge Exploiting lexical resources, such as dictionaries and thesauri, for the enrichment of the ontologies or as basis for a similarity measure. Ontology enrichment entails that additional information is added to each concept's descriptions by searching for relevant information in the resource (Montiel-Ponsoda *et al.*, 2011). The intent behind this approach is that existing similarity measures are likely to perform better if more information is available. A resource can also be used as basis for a similarity measure

[10] Quick-Ontology-Mapping (QOM).

[11] **G**eneric **O**ntology **M**apping and **M**apping M**a**nagement (GOMMA).

(Budanitsky and Hirst, 2001; Budanitsky and Hirst, 2006). This can be done by allocating an appropriate entry for each concept within the resource. The similarity between two concepts can then be determined by comparing their corresponding lexical entries, e.g. by computing the textual overlap between their definitions.

Ontology-based background knowledge Exploiting ontologies as background knowledge. These ontologies can be domain specific (Aleksovski, 2008), upper-level descriptions (Niles and Pease, 2001; Matuszek *et al.*, 2006) or automatically retrieved from the semantic-web (Sabou, d'Aquin, and Motta, 2008). Similarly to the previous category, an equivalent concept is identified in the background ontology for each concept in the given ontologies. From here, one can enrich the given ontologies using the information of the background ontology, compute semantic similarities between concepts by analysing their distances within the background ontology or even infer mappings.

1.3.3 Automatic Configuration

Ontology mapping systems tend to be quite complex, requiring algorithms for the computation, aggregation and processing of concept similarities and the extraction of correspondences. Many ontology mapping systems have emerged over the years, as evidenced by the amount of participating systems in the most recent Ontology Alignment Evaluation Initiative (OAEI) (Grau *et al.*, 2013). These systems are quite diverse with regard to their structure and applied techniques, though not one single system is able to perform exceptionally well on all data sets. Based on this, it is reasonable to assume that no single set-up of a mapping system will perform exceptionally in all circumstances. The issue of automatic configuration is also of important in the field of information retrieval, where the specific configuration of a retrieval system can have a large impact on the output quality (Oard *et al.*, 2008).

To overcome this limitation, it is necessary for a system to adapt itself to be better suited to solve the given mapping problem. The challenge here is to develop approaches which tackle three distinct tasks: (1) component selection, (2) component combination and (3) component tuning. The term component can be interpreted both as similarity measure and mapping system. Within a single mapping system, the process of configuration entails which similarities are selected for a given matching task, what approaches are used to combine their results and what parameters are used for the similarity metrics and combination approaches. For a meta-system, i.e. a mapping system by utilizing pre-existing mapping system and combining their results, the process of configuration entails which mapping systems are selected for a given task, how their results are combined and what parameters are used for each individual system. In general, in order create an appropriate configuration with the available components it is necessary to analyse certain properties of the given ontologies, for instance by evaluating their sizes or the richness of available meta-information. Based on this analysis one can choose the most appropriate selection and configuration of the available components (Mochol and Jentzsch, 2008; Cruz *et al.*, 2012; Lee *et al.*, 2007).

1.3.4 User involvement

For corporate-based applications of ontology mapping, notably schema-integration, the results of a mapping system are typically inspected and repaired or approved by a domain expert. Depending on the domain the given mapping problem can be very large, rendering it particularly difficult and time consuming for a domain expert to thoroughly inspect the entire result alignment. However, in addition to the domain expert(s) another source of human capital is likely available: the user. The core challenge of *user involvement* is to create techniques tailed to the expertise of the user such these can participate in the mapping process. This can be a particularly challenging task for dynamic applications where the user issues queries in natural language, and hence cannot be expected to be mapping specialist or domain expert.

One way to involve the user is to create a graphic visualization of the generated alignment, allowing the user to validate alignments by quickly and intuitively browsing, inspecting and modifying elements of the alignment (Mocan, Cimpian, and Kerrigan, 2006; Falconer and Storey, 2007; Raffio *et al.*, 2008). Another approach is to involve the user earlier in the mapping process. The *PROMPT* tool is an example of such a solution (Noy and Musen, 2003). The tool repeatedly presents mapping suggestions to the user and records their responses. Using these responses the system's beliefs about the computed concept similarities are updated. Subsequently, the system attempts to find inconsistencies and potential problems, which can in turn be presented to the user as new prompts until no further issues are detected.

Alternatively, the user can be involved in a process prior to matching, where the result of that process can be used as additional background knowledge. An example of this is the *HAMSTER* tool (Nandi and Bernstein, 2009), which gathers click-logs of a search-engine in a database. The information within this database is then used as a basis for an additional similarity measure.

1.3.5 Correspondence justification

Typically, ontology mapping systems annotate each correspondence with a confidence value, signifying the system's degree of confidence that the given correspondence is true. This value is typically defined to be in the range of $[0, 1]$. However, what a particular value signifies is open to interpretation since there is not always additional information on how the system derived a particular value. This issue is further elaborated in Section 3.1.2. In order to encourage the widespread acceptance of ontology mapping systems, it will become necessary that each correspondence of an alignment is also annotated with an explanation. Justified mapping results would enable a user to understand the behaviour of the system better, increasing the user's satisfaction. A justification would need to satisfy several criteria. First, it must be intuitive to a human user, for instance by using a visual representation. Second, it should also be available in machine-readable form, since systems which utilize alignments might also exploit the justification behind the correspondences.

One of the first attempts at providing users extended justifications for mappings can be seen in the S-MATCH system (Shvaiko *et al.*, 2005). S-MATCH can provide concise explanations, which can be extended with the use of background knowledge or logical reasoning and visually presented to the user. Matchers which employ agent-based argumentation systems to derive their mappings can also use the same argumentation for the eventual formulation of justifications aimed at users (Trojahn *et al.*, 2011).

1.3.6 Crowdsourcing

A new approach to large-scale problem solving, facilitated by the availability of the internet and social networks, is crowdsourcing (Brabham, 2008). Crowdsourcing relies on a large group of humans collaborating in the task of solving a particular problem. A given problem needs to presented in such a way that a common user can participate in creating a solution. Improving the usability of the crowdsourcing tool improves its performance, since a lower bar of entry allows for a larger userbase to participate. Thus, the core challenge is to devise methods allowing a general userbase to participate in the mapping process, generating high quality alignments from the user data and reconciling inconsistencies which may occur from contradicting user input.

Crowdsourcing has initially been applied as a means to establishing ontology interoperability in the work of Zhdanova and Shvaiko (2006). The produced tool allows user to generate, modify, store, share and annotate ontology mappings. However, the collaborative aspect is only sequential in nature, meaning that collaborative matching only occurs if a user decides to modify an already existing alignment. The tool is not moderated and lacks capabilities to resolve user disagreements. Disagreements must be resolved by the users themselves by voicing their opinions in the mapping annotations. A more user-friendly approach is to present the user small sub-problems of the mapping task, formulated as a user friendly questions (McCann, Shen, and Doan, 2008). This allows for a multitude of users to concurrently work on the same mapping problem. The user responses are gathered such that a mapping can be derived. For each unique user prompt, the majority answer of the users is accepted if the gap to the minority answer is bigger than a statistically significant margin. Instead of crowdsourcing the generation of correspondences, one can involve the user in the process of verifying correspondences and resolving inconsistencies. Here, a contemporary system would provide the correspondences such that the reconciliation task is formulated as a crowdsourcing project, with the users providing their insight how detected inconsistencies should be resolved (Nguyen *et al.*, 2013).

1.3.7 Alignment infrastructures

As seen from the previously listed challenges and applications, many aspects in the area of ontology mapping rely on the availability and management of ontologies. For instance, to facilitate collaborative matching, as described in subsection 1.3.6, an entire support system needs to be created which facilitates this collaboration. The core challenge here is the development of a support system which enables users and other knowledge systems to perform alignment related tasks (Euzenat, 2005). The most prominent tasks are: (1) alignment storage, (2) alignment retrieval, (3) alignment computation, either manually or using a supplied mapping system, (4) alignment revision and (5) alignment-based information translation.

There has been some work done with regard to this challenge. Some systems focus on a particular task of an infrastructure (Noy and Musen, 2003; Noy, Griffith, and Musen, 2008), though these system can act as modules of a complete infrastructure. An example of an initial attempt at constructing a complete infrastructure is the alignment server (Euzenat, 2005), where users and system can interact with the system through the use of the alignment API (Euzenat, 2004a). Another example of such an infrastructure is the *Cupboard* system (d'Aquin and Lewen, 2009). It facilitates functionalities such as alignment storage and retrieval, though lacks the ability to compute alignments or translate information using an alignment.

1.4 Problem Statement and Research Questions

In the previous sections we introduced the process of ontology mapping as a solution for resolving the interoperability between heterogeneous knowledge systems, and discussed numerous applications for which ontology mapping is of importance. Furthermore, we highlighted several key areas of interest which have been established as future focus points of research. This thesis focuses on one of these key areas, being ontology mapping with background knowledge. The aim of these techniques is to exploit sources of information other than the available meta-data of the ontologies. The following problem statement will guide the research:

> **Problem statement** *How can we improve ontology mapping systems by exploiting auxiliary information?*

We identify two main types of auxiliary information sources which can be used to enrich ontology mapping systems, being *lexical resources* and *partial alignments.* Using this categorization we identify four research questions which guide the research with regard to the problem statement. The questions address the problems of (1) accurately linking ontology concepts to lexical senses, (2) exploiting partial alignments to derive concept similarities, (3) evaluating correspondences of partial alignments and (4) matching ontologies with little to no terminological overlap using partial alignments.

> **Research question 1** *How can lexical sense definitions be accurately linked to ontology concepts?*

A lexical resource is a corpus containing word definitions in terms of senses, synonyms and semantic relations that hold between senses. By linking ontology concepts to word senses of a lexical resource, one can exploit the meta-data of the lexical resource in order to determine how semantically similar two ontology concepts are. For example, one can determine the similarity between two senses by inspecting their locations within the taxonomy of the lexical resource. Their similarity can be established by computing the distance between the two senses, i.e. the length of the shortest path between the two senses, using graph-theory-based measures. If there is a low semantic distance between two given senses, then the implication is that there is a strong semantic relation between theses senses, meaning that ontology concepts which are associated with these senses are likely to encode the same information. However, the correctness of this semantic distance strongly relies on the ability to identify the correct sense for each ontology concept. Words are often ambiguous, leading to the problem that several lexical senses can match a ontology concept label. At this point one needs to discard senses which do not accurately represent the intended meaning of a given concept. Our research aims at extending current methods of determining concept senses.

> **Research question 2** *How can we exploit partial alignments in order to derive concept correspondences?*

Businesses will often employ domain experts to handle mapping problems of big schema integration tasks. The knowledge of the domain expert ensures that all produced correspondences are indeed correct, ensure maximum interoperability between two knowledge systems. However, mapping large ontologies is a very laborious task. Ontologies of large systems can contain several thousand concepts, meaning that a mapping between two large ontologies can consist of several thousand correspondences. It must be ensured that the produced correspondences are correct and logically consistent. The scale of such a task can easily be too much for a domain expert. One branch of research which attempts to tackle this issue involves the creation of tools which reduce the workload of the domain expert (Noy and Musen, 2003; Cruz, Sunna, and Chaudhry, 2004). Such tools aide the expert by for instance intuitively visualizing the ontologies and their mappings, suggesting new correspondences and performing consistency checks and logical inference between the ontologies. However, a domain expert might not be willing to invest time into the familiarization with a mapping tool and possible abort the task because it is deemed too daunting. Alternatively, an expert might only available for a certain amount of time, enabling him to only generate a small amount of correspondences. In these situations, it is often the case that an incomplete alignment is produced which needs to be completed. This incomplete alignment, also referred to as partial alignment, can be a valuable source of information when determining the remaining correspondences. Our research aims at developing a novel approach at utilizing existing partial alignments in order to determine concept similarities.

> **Research question 3** *How can we evaluate whether partial alignment correspondences are reliable?*

Methods which exploits partial alignments for the sake of finding the remaining correspondences rely on the correctness of the exploited correspondences from the partial alignment. The performance of these methods is affected significantly depending on the amount of incorrect correspondences within the partial alignment. To ensure that these specialized methods perform as designed, one must be evaluate the provided correspondences to ascertain whether these are indeed correct. Typically, such evaluations involve similarity-based methods, which measure the overlap of meta-data of ontology concept definitions. Our research attempts to complement these methods by determining the consistency of the input correspondences with regard to a set of reliably generated correspondences.

Research question 4 *To what extent can partial alignments be used in order to bridge a large terminological gap between ontologies?*

A particular type of heterogeneity that can exist between two equivalent concepts x and y is a terminological heterogeneity. This describes a situation in which x and y are defined using significantly different labels and annotations. This excludes minor character differences, e.g. the differences between the labels *'Accepted_Paper'*, *'Accept-Paper'* and *'Paper (Accepted)'*. The challenging situation here is that many elemental similarities cannot derive an appropriate similarity values between x and y. Ontology mapping systems can avoid this issue by analysing the similarities between related concepts of x and y, by for instance comparing the labels of the parents of x and y, though if the related concepts of x and y are also heterogeneous then this no longer is a feasible approach. We say that there is a *terminological gap* between ontologies O_1 and O_2 if there is little to no terminological overlap between O_1 and O_2. This represents a challenging matching scenario where specialized approaches are required in order to succeed. A common approach here is to extract additional concept term from a lexical resource, thus increasing the likelihood that x and y will contain similar labels. This approach requires that an appropriate lexical resource is available for each matching domain, meaning that it is ineffective if no appropriate resource exists for a given matching task. However, it might be that a partial alignment is available for the given task. Our research aims at developing an approach for mapping terminological heterogeneous ontologies by utilizing partial alignments.

1.5 Thesis Overview

The remainder of this thesis is structured as follows. Chapter 2 provides the reader with a formal definition of the task of ontology matching, while also introducing methods that are applicable for the evaluation of alignments. Chapter 3 provides an overview of existing mapping approaches. Here we provide an introduction to mapping system architectures, the three core tasks which a mapping system needs to perform and an overview of approaches for each of the three core tasks. We conclude this chapter by providing a survey of state-of-the-art mapping systems.

Chapter 4 answers the first research question. We introduce a method utilizing virtual documents for measuring the similarity between ontology concepts and sense definitions and define a framework which links ontology concepts to lexical senses. Chapter 5 addresses the second research question. We propose an approach which measures the similarities between a given concept and the given correspondences of the partial alignment, also referred to as anchors, which are compiled into a *anchor-profile*. Two concepts are matched if their respective anchor-profiles are similar. The third research question is addressed in Chapter 6. Our approach aims at reformulating the anchor-evaluation problem as a feature-evaluation task, where every anchor is represented as a feature. Chapter 7 answers the fourth research question. Our approach aims to enrich the concept profiles each ontology with the terminology of the other ontology by exploring the semantic relations which are asserted in the partial alignments. In Chapter 8 we summarize the contributions to each research question and identify promising directions of future work.

Chapter 2

Background

Ontology mapping is the essential process facilitating the exchange of information between heterogeneous data sources. Here, each source utilizes a different ontology to model its data, which can lead to differences with respect to the syntax of the ontology, concept naming, concept structuring and the granularity with which the knowledge domain is modelled. Euzenat (2001) identified three main heterogeneity categories as terminological, conceptual and semiotic heterogeneities. Given two ontologies, these heterogeneities need to be resolved, which in turn allows for the exchange of information between any knowledge system which uses any of the two given ontologies to model its data. This is achieved by mapping concepts of one ontology to the concepts of the other ontology which model the same data. The mappings are then compiled into a list of correspondences, referred to as an *alignment*.

As an example, suppose that the European Commission (EC) would start an initiative to centralize all vehicle registrations over all member countries. Then, the EC would have to create a central knowledge system ontology that is designed to cover the combination of all requirements of the member countries. To fully realize this integration effort, every country would need to create a mapping between the ontology of its own registration system and the new ontology of the central EC system. Furthermore, it might be the case that a given country manages multiple registration systems. These might handle vehicle types separately, e.g. cars, trucks and motorcycles, or model public, government and military vehicles separately. As a result, a country would have to make several mappings in order to transfer the data of every knowledge system. Figure 2.1 displays two example ontologies which can be used to model vehicle registrations and an example mapping between the two ontologies.

One can see that some concepts are straight-forward to match since they model the same entity and have similar names, as for instance *Car-Car* and *Bike-Bicycle*. However, other corresponding concepts do have more pronounced differences. *Vehicle* and *Conveyance* model the same top-level concept, though exhibit no syntactic overlap due to the use of synonyms. Concepts which do no exhibit a significant overlap of meta-information are typically harder to match, requiring the usage of more advanced techniques, e.g. Natural-Language Processing, or the exploitation of

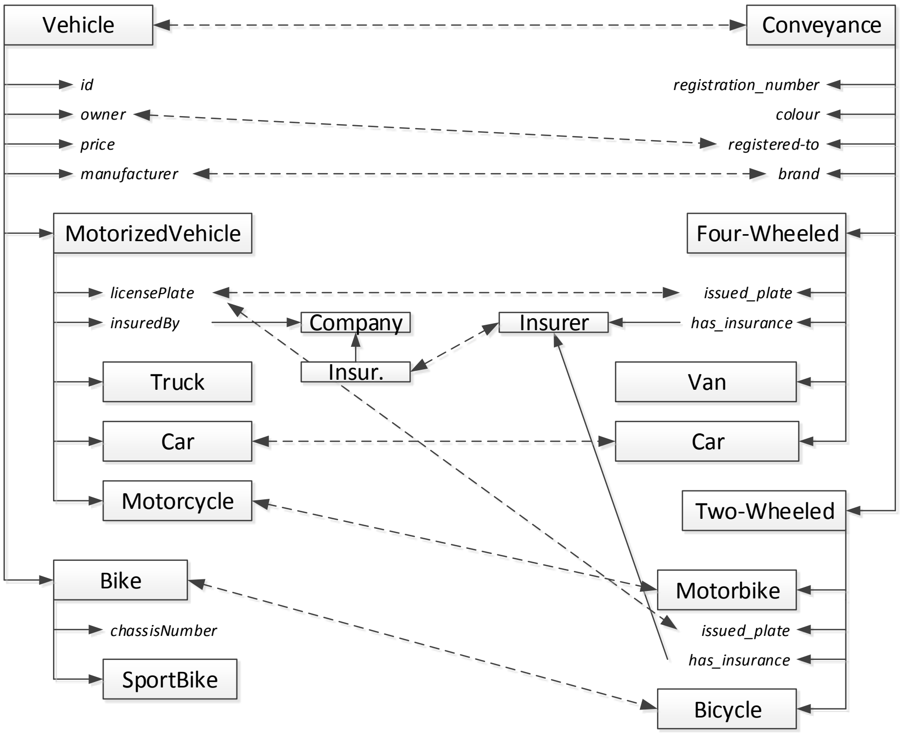


Figure 2.1: Example mapping between two small ontologies.

a broader rage of information.

The example of Figure 2.1 displays correspondences between concepts which can be interpreted as equivalent. Identifying correspondences of this type significantly benefits the facilitation of information exchange. If for a given concept a equivalent concept has been identified, then the only operation that still needs to be performed for that concept is the generation of a transformation function. This function can express any instance of one concept using the terminology of the other concept's ontology. If for a given concept an equivalent concept cannot be located, it is still possible to facilitate information exchange by for instance using the transformation function of a parent concept for which an equivalent correspondences has been identified. Alternatively, one can identify other semantic relations between concepts, e.g. *generalization* or *overlapping*, in order to help identify possible target concepts for the transformation functions. For practical purposes, each correspondence is typically annotated with a degree of confidence in the interval of $[0, 1]$. This measure expresses the amount of trust one has in the truthfulness of that correspondence. It is typically based on the results of several algorithms measuring the overlap of meta-data between the concepts. Note that this measure of trust is not equivalent to the probability of the given correspondence being correct. where a value of 0.7 would mean that the expected outcome of sampling 10 correspondences with trust value 0.7 would be 7 correct and 3 incorrect correspondences. This topic will be further discussed in Section 3.1.2. We have updated the example in Figure 2.1 with different types of relations and some confidence measures, shown in Figure 2.2.

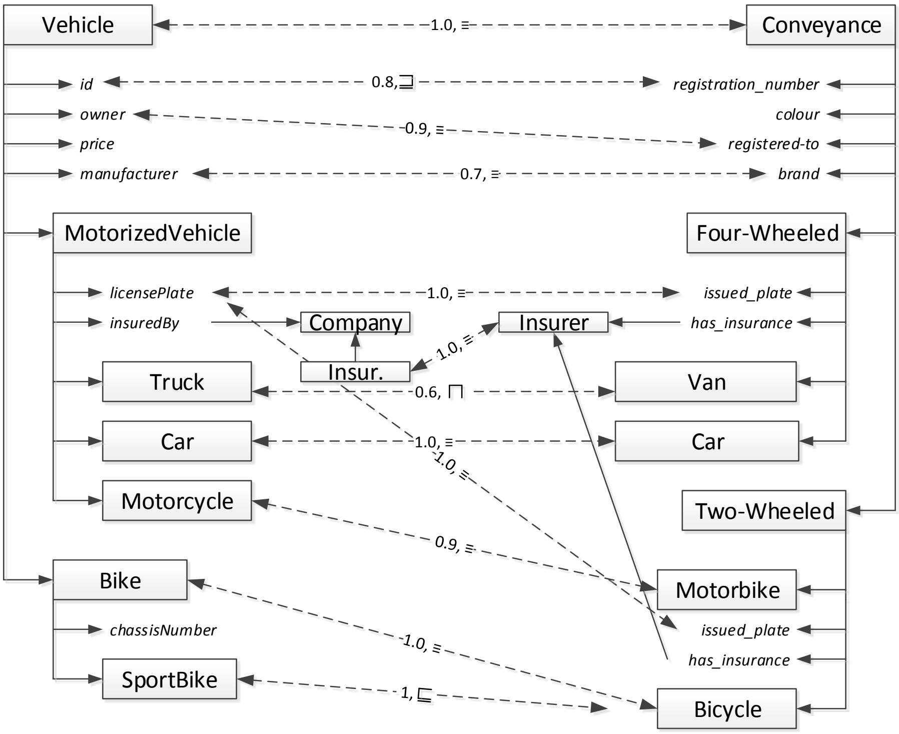


Figure 2.2: Example mapping between two small ontologies. The mapping models different semantic relation types and includes confidence values for each correspondence.

In Figure 2.2, we see that all correspondences that were also depicted in Figure 2.1 have been annotated using the *equivalence* (≡) symbol. New correspondences expressing different types of semantic relations are also depicted. One such correspondence is the connection between *SportsBike* and *Bike*. These two concepts are not equivalent, however a sports-bike is certainly a type of bike. Thus the correspondence is annotated using the *generalization* (⊒) relation. Another generalization can be seen in the correspondence between *id* and *registration_number*. This correspondence notably has a slightly lower confidence value. The correspondence between *Truck* and *Van* is annotated with a overlapping (⊓) relation type, indicating that while these two concepts can contain common instances, they cannot be associated using a more precise relation type, such as generalization or equivalence.

2.1 The Mapping Problem

We define ontology mapping as a process which takes as minimal input two ontologies O_1 and O_2 and produces an output alignment A'. Furthermore, this process can take as input an already existing alignment A, a set of external resources r and a set of parameters p. The pre-existing alignment can originate from a different system, thus allowing the combination of two systems in a cascade arrangement, or from the same system, allowing the possibility of designing an iterative mapping process. The set of parameters p incorporates any parameter which influences the mapping process, such as settings, weights or thresholds. While r is broadly defined, in practice the most commonly used resources are linguistic or domain thesauri and auxiliary ontologies.

To formally establish the mapping problem, we start by defining an ontology as follows:

Definition 1 (**Ontology**). *An ontology O is a tuple $O =< C, P, I, T, V, R >$, such that:*
C is a set of classes;
P is a set of properties;
I is a set of instances;
T is a set of data-types;
V is a set of data-values;
R is a specific set of relations modelled by the ontology language.

This definition encapsulates the types of entities that are typically modelled in an ontology. Note that the entities contained in C and P can be both referred to as *concepts* in order to conveniently refer to the entities that are matched in a mapping scenario. Ontology languages such as *OWL* facilitate the creation of such entities using specially defined constructs:

Classes The primary concepts of an ontology. These form the conceptualization of a domain and can be interpreted as sets of instances. For example, the concepts *Car* and *Bike* from Figure 2.1 are classes.

Properties Essential to the storage of data, properties model the relation between instances and specific data-values, or the relation between instances and other instances. The two types of properties are expressed in *OWL* as `owl:DataProperty` and `owl:ObjectProperty`. As an example, the property *insuredBy* from Figure 2.1 is used to connect instances of the class *MotorizedVehicle* and the class *Insurance*, whereas the property *licensePlate* would connect individual cars with data-values corresponding to their license plate.

Instances Individual instantiations of classes in the ontology consisting of a series of associated data values, related instances and references to the corresponding class that is being instantiated. This is equivalent to a row of data values in a table of a database. In *OWL*, instances are expressed under the `owl:Thing` construct, where the classes being instantiated are referred to using the `rdf:type` construct. While in practice instances are typically stored in a different file than the corresponding ontology definition, they are still considered to be part of the ontology (Ehrig, 2006; Euzenat and Shvaiko, 2007).

Data-Types A classification of the various types of data. A data-type specifies all possible values that can be modelled, which operations can be performed on this data, the ways values of this data-type can be stored and optionally the meaning of the data. For example `string`, `integer` and `xsd:dateTime` are data-types. `string` models all possible combinations of characters, `integer` only models whole numbers and `xsd:dateTime` models specific time stamps according to a specified syntax, thus also providing a meaning to the data.

Data-Values Simple values that fall into the domain of a specified data-type. As an example, the name and contact information of a specific vehicle owner are typically stored as values.

Relations The set of relations already modelled in the given ontology language. This set is shared over every ontology that is modelled using the given language and fundamental in the construction of every individual ontology. An ontology includes at least the following relations:

- *specialization* ($\sqsubseteq$), defined on $(C \times C) \cup (P \times P) \cup (I \times I)$
- *disjointness* ($\perp$), defined on $(C \times C) \cup (P \times P) \cup (I \times I)$
- *assignment* ($=$), defined on $I \times P \times (I \cup V)$
- *instantiation* ($\in$), defined on $(I \times C) \cup (V \times T)$

Examples of additional relations which an ontology language might model are *overlapping* ($\sqcap$) or *part-of* ($\subset$). For non-symmetric relations a language might also model their inverse relations, such as *generalization* ($\sqsupseteq$), being the inverse of *specialization*, and *consist-of* ($\supset$), being the inverse of *part-of*.

The ultimate goal of ontology mapping is to find a way that allows us to alter instances of one ontology such that they conform to the defined structure and terminology of the other. To achieve this for a given instance, one must identify to which class of the other ontology the instance should belong to, what data can be transferred by allocating matching properties and how the data-values must be altered such that these conform to other data-types. Any mapping system must therefore be able to identify corresponding classes and properties such that these transformation rules can be generated.

An identification of a relation between two entities, e.g. matching classes or properties, is expressed as a *correspondence*. We define a *correspondence* as follows:

Definition 2 (**Correspondence**). *A correspondence between two ontologies O_1 and O_2 is a 5-tuple $< id, e_1, e_2, t, c >$, where:*

id is a unique identifier allowing the referral to specific correspondences;

$e_1 \in O_1$ is a reference to an entity originating from the first ontology;
$e_2 \in O_2$ is a reference to an entity originating from the second ontology;
t denotes the relation type between e_1 and e_2;
c is a confidence value in the interval $[0, 1]$.

Thus, a given correspondence $< id, e_1, e_2, t, c >$ asserts that a relation of the type t holds between entities e_1 and e_2 with a degree of confidence c. e_1 and e_2 are typically modelled as a URI in order to refer to a specific entity. Relation types which are typically asserted in a correspondence are generalization ($\sqsupseteq$), specialization ($\sqsubseteq$), disjointness ($\perp$), overlapping ($\sqcap$) and equivalence ($\equiv$).

As an example, we can express one of the correspondences displayed in Figure 2.2 as follows:

$$< \text{id123}, \textit{SportBike}, \textit{Bike}, \sqsubseteq, 1.0 >$$

This correspondence asserts that the class *SportBike* is a subclass of *Bike* with a confidence of 1.0, meaning that any instance of *SportBike* is also an instance of *Bike*.

Note that the entities of a correspondence can also refer to properties or instances. Thus, the correspondence between the properties *owner* and *registered_to* from Figure 2.2 can be expressed as follows:

$$< \text{id124}, \textit{owner}, \textit{registered\_to}, \equiv, 0.9 >$$

The ultimate goal of the ontology mapping process between two ontologies is to identify all appropriate correspondences. These correspondences are gathered into a set, which is referred to as an *Alignment* or *Mapping*. Formally, we define an alignment between two ontologies as follows:

Definition 3 (**Alignment**). *An alignment A between two given ontologies O_1 and O_2 is a set of correspondences $A = \{c_1, c_2, \dots c_n\}$, such that for each correspondence $< id, e_1, e_2, t, c > \in A$, $e_1 \in O_1$ and $e_2 \in O_2$ holds.*

The example in Figure 2.2 illustrates a possible mapping between two ontologies. The correspondences in this example can be expressed as an alignment A as follows:

$$A = \left\{ \begin{array}{r} < \text{id1}, \textit{Vehicle}, \textit{Conveyance}, \equiv, 1.0 > \\ < \text{id2}, \textit{id}, \textit{registration\_number}, \sqsupseteq, 0.8 > \\ < \text{id3}, \textit{owner}, \textit{registered-to}, \equiv, 0.9 > \\ < \text{id4}, \textit{manufacturer}, \textit{brand}, \equiv, 0.7 > \\ < \text{id5}, \textit{licensePlate}, \textit{issued\_plate}, \equiv, 1.0 > \\ < \text{id6}, \textit{Insur.}, \textit{Insurer}, \equiv, 1.0 > \\ < \text{id7}, \textit{Truck}, \textit{Van}, \cap, 0.6 > \\ < \text{id8}, \textit{Car}, \textit{Car}, \equiv, 1.0 > \\ < \text{id9}, \textit{Motorcycle}, \textit{Motorbike}, \equiv, 0.9 > \\ < \text{id10}, \textit{Bike}, \textit{Bicycle}, \equiv, 1.0 > \\ < \text{id11}, \textit{SportBike}, \textit{Bicycle}, \sqsubseteq, 1.0 > \end{array} \right\}$$

A mapping system typically exploits all available information in order to produce an alignment. At the very least, this includes the meta-information that is encoded into the ontologies themselves. This information can take shape in the form of concept labels, descriptions or annotations, relations between entities and given instantiations of concepts. However, there exists a range of information which can be supplied to the mapping system as additional input. This range can be categorized as follows:

Input Alignments It can be the case that an alignment between the two given ontologies is available. This alignment may be supplied by another mapping framework, available from a repository or the result of a domain expert attempting to map the given ontologies. In this case, the correspondences of that alignment can be used to guide the mapping process.

Parameters Ontology Mapping systems tend to be quite complex, which is necessary in order to deal with all possible types of input ontologies. These systems typically require a set of parameters that fine-tune all possible aspects of the systems to maximize performance. Typical forms of parameters are thresholds, function parameters or system specifications. Note that the core challenge in section 1.3.3 envisions the elimination of this type of input, where a mapping is able to derive this parameter set autonomously.

Knowledge Resources Any type if information that is not associated with the given mapping problem or the applied mapping system are categorized as knowledge resource. Typical resources which can be exploited are domain ontologies, lexical resources or internet-based resources.

Given the definition of an ontology in definition 1, the definition of an alignment in definition 3 and the different types of additional inputs that a mapping system could exploit, we can formally define the ontology mapping process as follows:

Definition 4 (Ontology Mapping). *Ontology Mapping is a function which takes as input a pair of ontologies O_1 and O_2, an alignment A, a set of resources r and a set of parameters p, and returns an alignment A' between the ontologies O_1 and O_2:*

$$A' = f(O_1, O_2, A, r, p)$$

Note that O_1 and O_2 are mandatory inputs, while A, r and p are optional, possibly requiring special techniques in order to adequately exploit the additional information. The process of ontology mapping is visualized in Figure 2.3:

When matching with *input alignments*, one can distinguish between three variants of this problem:

Alignment Refinement In this variant a (nearly) completed alignment is available as input. The main task here does not involve the discovery of new correspondences, but rather the refinement of the existing ones. For example, a typical approach that is applied for this type of problem is consistency checking through the application of reasoning techniques and resolving the resulting conflicts.

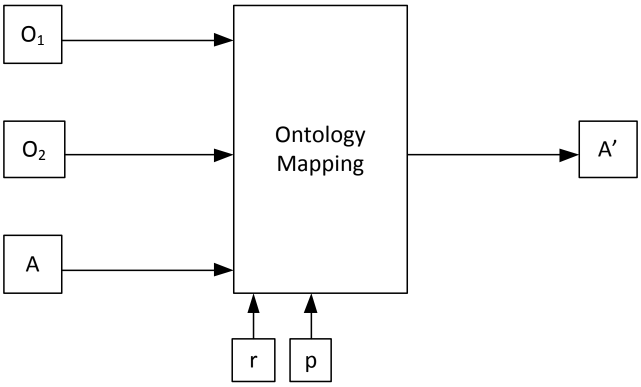


Figure 2.3: Visualization of the ontology mapping process

Alignment with Tertiary Ontologies When given two ontologies O_1 and O_2, it might be possible that existing alignments link either ontology to one or more ontologies which are not part of the original matching problem. If a chain of alignments exists which link O_1 and O_2 through one or more tertiary ontologies, then techniques such as logical inference can be applied in order to infer correspondences. Otherwise, the alignments must be exploited differently. For instance, one can use the information of these ontologies as additional context, similar to the core challenge described in section 1.3.2. Alternatively, one can shift the mapping task to one or more of the tertiary ontologies if one can identify a mapping task which is easier to solve.

Partial Alignments In some situations it may the case that an attempt to create an alignment was started but aborted unfinished. A prime example of this is a domain expert being unable to finish an alignment due to time constraints. In this situation the task here is to complete the alignment. The challenge here is to find ways in which the existing correspondences can be exploited in order to determine the remaining correspondences.

The correspondences of an input alignment are also formulated as 5-tuples, as defined in Definition 2. Additionally, the correspondences of a partial alignment (PA) are typically referred to as *anchors* in order to clearly separate these from other types of correspondences, such as the correspondences of the result or reference alignment.

Definition 5 (Anchor). *Given two ontologies O_1 and O_2, and a given partial alignment PA between O_1 and O_2, an anchor $\boldsymbol{a}$ is defined as a correspondence such that $\boldsymbol{a} \in PA$.*

Having formally introduced the ontology mapping process, along with the alignment as desired output, it becomes necessary to be able to analyse a given alignment with respect to its quality, such that the effectiveness of a given mapping approach can be quantitatively established. We will introduce the methodology of alignment evaluation in the next subsection.

2.2 Evaluation of Alignments

As elaborated in section 2.1, the goal of an ontology mapping system is to produce an alignment which facilitates the transfer of information between two ontologies. In order to evaluate the quality of a proposed mapping system or approach, there must be a quantitative way to evaluate alignments between ontologies.

The most common way to evaluate an alignment is through the application of a golden standard. Here, a domain expert creates a reference alignment which represents the ideal outcome when mapping the two given ontologies. We will illustrate this process using a running example using the ontologies depicted in Figure 2.1. Let us assume we are given the following reference alignment R:

$$R = \left\{ \begin{array}{r} < \mathrm{id1}, \mathit{Vehicle}, \mathit{Conveyance}, \equiv, 1.0 > \\ < \mathrm{id2}, \mathit{id}, \mathit{registration\_number}, \sqsupseteq, 1.0 > \\ < \mathrm{id3}, \mathit{owner}, \mathit{registered\text{-}to}, \equiv, 1.0 > \\ < \mathrm{id4}, \mathit{manufacturer}, \mathit{brand}, \equiv, 1.0 > \\ < \mathrm{id5}, \mathit{licensePlate}, \mathit{issued\_plate}, \equiv, 1.0 > \\ < \mathrm{id6}, \mathit{Insur.}, \mathit{Insurer}, \equiv, 1.0 > \\ < \mathrm{id7}, \mathit{Truck}, \mathit{Van}, \cap, 1.0 > \\ < \mathrm{id8}, \mathit{Car}, \mathit{Car}, \equiv, 1.0 > \\ < \mathrm{id9}, \mathit{Motorcycle}, \mathit{Motorbike}, \equiv, 1.0 > \\ < \mathrm{id10}, \mathit{Bike}, \mathit{Bicycle}, \equiv, 1.0 > \\ < \mathrm{id11}, \mathit{SportBike}, \mathit{Bicycle}, \sqsubseteq, 1.0 > \end{array} \right\}$$

This alignment corresponds to the alignment depicted in Figure 2.2, with the alternation that all correspondences have a set confidence of 1.0.

Next, let us assume that a hypothetical ontology mapping system produces the following output alignment A:

$$A = \left\{ \begin{array}{r} <\text{id1}, \textit{Vehicle}, \textit{Conveyance}, \equiv, 0.8> \\ <\text{id2}, \textit{id}, \textit{registration\_number}, \sqsupseteq, 0.5> \\ <\text{id5}, \textit{licensePlate}, \textit{issued\_plate}, \equiv, 0.9> \\ <\text{id6}, \textit{Insur.}, \textit{Insurer}, \equiv, 1.0> \\ <\text{id8}, \textit{Car}, \textit{Car}, \equiv, 1.0> \\ <\text{id9}, \textit{Motorcycle}, \textit{Motorbike}, \equiv, 1.0> \\ <\text{id10}, \textit{Bike}, \textit{Bicycle}, \equiv, 0.75> \\ <\text{id12}, \textit{Truck}, \textit{Car}, \sqsubseteq, 0.7> \\ <\text{id13}, \textit{MotorizedVehicle}, \textit{Four-Wheeled}, \equiv, 0.7> \\ <\text{id14}, \textit{chassisNumber}, \textit{issued\_plate}, \equiv, 0.6> \end{array} \right\}$$

The question now is what measures can be applied in order to compare A with R. For this purpose one can use the measures of *Precision* and *Recall*, which stem from the field of information retrieval (Rijsbergen, 1979). These measure the 'correctness' and 'completeness' of a set with respect to another set. Given an alignment A and a reference alignment R, the precision $P(A, R)$ of alignment A can be calculated as follows:

$$P(A, R) = \frac{|R \cap A|}{|A|} \tag{2.1}$$

The recall $R(A, R)$ of alignment A can be calculated as follows:

$$R(A, R) = \frac{|R \cap A|}{|R|} \tag{2.2}$$

The output alignment and reference alignment of our running example are visualized as sets in Figure 2.4, where each correspondence its represented by its identification value:

We can see in Figure 2.4 an emphasized area, which corresponds to the overlapping area between A and R. The implication here is that correspondence in this area have been correctly identified by the given ontology mapping system. Additionally, some correspondences are located only in R, meaning that the system failed to identify these correspondences. On the other hand, the correspondences which are only in A are incorrect and erroneously included in the output alignment.

Using the measures of precision and recall, we can now evaluate the quality of our example alignment as follows:

$$P(A, R) = \frac{|R \cap A|}{|A|} = \frac{|\{\textit{id1,id2,id5,id6,id8,id9,id10}\}|}{|\{\textit{id1,id2,id5,id6,id8,id9,id10,id12,id13,id14}\}|} = \frac{7}{10} = 0.7$$

$$R(A, R) = \frac{|R \cap A|}{|R|} = \frac{|\{\textit{id1,id2,id5,id6,id8,id9,id10}\}|}{|\{\textit{id1,id2,id3,id4,id5,id6,id7,id8,id9,id10,id11}\}|} = \frac{7}{11} = 0.6363$$

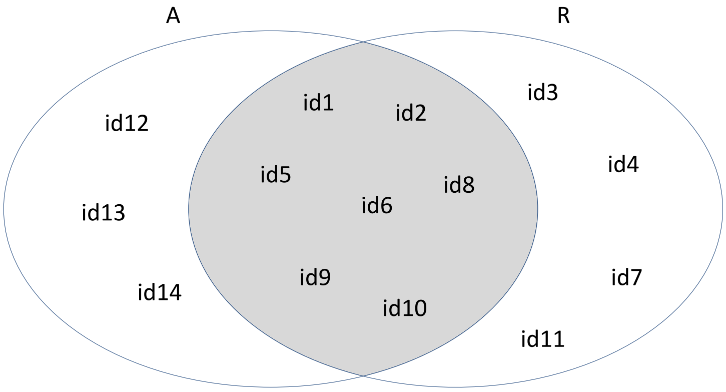


Figure 2.4: Visualization of the interaction between the example alignment and the example reference.

We can distinguish all possible correspondences when viewing the mapping problem as a binary classification task, where one must individually classify all possible correspondences as either *true* or*false*. The implication here is that all correspondences which are classified as *false* are simply not included in the output alignment. The error matrix of such a task is as follows:

	Classified True	Classified False
Actually True	TP	FN
Actually False	FP	TN

The set of correspondences is partitioned into: *true positive (TP)*, *false positive (FP)*, *false negative (FN)* and *true negative (TN)* with respect to the desired classification and actual classification. Using these, we can derive the measures of *precision* and *recall* as follows:

$$P(A,R) = \frac{|TP|}{|TP + FP|} \tag{2.3}$$

$$R(A,R) = \frac{|TP|}{|TP + FN|} \tag{2.4}$$

When analysing the performance of ontology mapping systems, it is desirable to be able to formulate the performance of a system using a single value. However, using either precision or recall alone does not lead to a fruitful comparison. For instance, one would not regard a system with a recall value of 1 as good if it simply returned every possible correspondence. Using both precision and recall can be difficult, since both values represent a trade-off that needs to be managed, where an attempt at an increase of one measure often comes to the detriment of the other. A solution to this problem is the application of a combination of the two measures, which considers both precision and recall equally. For this purpose the *F-Measure* is typically deployed (Giunchiglia *et al.*, 2009). The F-Measure is defined in the interval of $[0, 1]$ and represents the harmonic mean between precision and recall. Given an alignment A, a reference alignment R, the precision $P(A, R)$ and recall $R(A, R)$, the F-Measure $F(A, R)$ of A with respect to R is defined as:

$$F(A,R) = \frac{2 \times P(A,R) \times R(A,R)}{P(A,R) + R(A,R)} \tag{2.5}$$

Returning to our running example, we can now express the quality of the example alignment as a single value using the F-Measure:

$$F(A,R) = \frac{2 \times P(A,R) \times R(A,R)}{P(A,R) + R(A,R)} = \frac{2 \times 0.7 \times 0.6363}{0.7 + 0.6363} = \frac{0.89}{1.3363} = 0.666$$

In most scenarios, it is desirable to consider both precision and recall equally when expressing the quality of an alignment as a single value. This may however not always be the case. For instance, a domain expert may wish to use an ontology mapping tool to generate a preliminary alignment, such that he can create a final alignment by verifying and altering the preliminary alignment. Let us assume that the expert has a choice between a set of mapping systems and he wishes to choose the system which will result in the least amount of work. One can argue that the expert would in this case prefer the system which tends to produce alignments with a significantly higher precision, since this would imply spending less time removing incorrect correspondences. However, the expert would also like some emphasis on the typical recall performance, since a nearly empty alignment with correct correspondences implies that he would have to perform most of the mapping duties manually anyway. Choosing a system using the measures that are introduced so far as performance indicator would not lead to a satisfactory result for the hypothetical domain expert. However, in Rijsbergen (1979) a generalized form of the F-Measure is introduced which allows for the weighting of either precision or recall to a specified degree.

Given an alignment A, a reference alignment R, the precision $P(A, R)$, the recall $R(A, R)$ and a weighting factor β, the weighted F-Measure is defined as follows:

$$F_\beta(A,R) = (1+\beta^2) \times \frac{P(A,R) \times R(A,R)}{(\beta^2 \times P(A,R)) + R(A,R)} \tag{2.6}$$

One can see that the weighted F-Measure is balanced when choosing $\beta = 1$. Thus, the F-Measure expressed in equation 2.5 is actually the weighted F_1 measure, though in the case of $\beta = 1$ the weight of β is typically omitted when referring to the F-Measure. Variants of β which are commonly used are 0.5 when emphasizing precision and 1.5 when emphasizing recall.

The introduced evaluation methods so far did not consider the confidence values of the produced correspondences. These confidence values typically impose a design challenge on the mapping system designers. After generating a set of correspondences, a system may apply a given threshold in order to dismiss correspondences which exhibit an insufficiently high degree of confidence. The choice of this threshold can heavily influence the resulting precision, recall and F-measure. A high threshold typically results in a high precision and low recall, whereas a low threshold typically results in a low precision and high recall. Judging the performance of a system by simply calculating the precision, recall and F-measure may result in an unrepresentative conclusion if a different and possibly higher performance could have been achieved by simply selecting a different threshold.

In order to circumvent the dilemma of selecting a specific threshold, one can apply a technique known as *thresholding.* We will illustrate this technique using our example alignment A. Table 2.1 lists all correspondences of A, sorted by their confidence values in descending order. The listed thresholds imply that all correspondences with a lower confidence value are discarded if that threshold were to be applied, whereas the F-measure listed next to a particular threshold means what the resulting F-measure would be after applying the threshold.

Correspondence	threshold	F-Measure
$<$ id6, *Insur.*, *Insurer*, $\equiv$, 1.0 $>$	1.0	0.428
$<$ id9, *Motorcycle*, *Motorbike*, $\equiv$, 1.0 $>$	1.0	0.428
$<$ id8, *Car*, *Car*, $\equiv$, 1.0 $>$	1.0	0.428
$<$ id5, *licensePlate*, *issued_plate*, $\equiv$, 0.9 $>$	0.9	0.533
$<$ id1, *Vehicle*, *Conveyance*, $\equiv$, 0.8 $>$	0.8	0.625
$<$ id10, *Bike*, *Bicycle*, $\equiv$, 0.75 $>$	**0.75**	**0.705**
$<$ id12, *Truck*, *Car*, $\sqsubseteq$, 0.7 $>$	0.7	0.631
$<$ id13, *MotorizedVehicle*, *Four-Wheeled*, $\equiv$, 0.7 $>$	0.7	0.631
$<$ id14, *chassisNumber*, *issued_plate*, $\equiv$, 0.6 $>$	0.6	0.6
$<$ id2, *id*, *registration_number*, $\sqsupseteq$, 0.5 $>$	0.5	0.666

Table 2.1: Sorted example correspondences with their respective thresholds and resulting F-measures.

We can see from Table 2.1 that a higher F-measure than 0.666 can be achieved if one were to apply a threshold of 0.75 before computing the F-measure. This threshold discards three incorrect and one correct correspondence, resulting in a F-measure of 0.705.

Using our example alignment, we have demonstrated the technique of *thresholding*, which is defined as applying the threshold which results in the maximum attainable F-measure (Euzenat *et al.*, 2010) [1] Let us define A_x as the resulting alignment after applying a threshold x to alignment A. We define the thresholded weighted F-Measure F^T_β as follows:

[1] Presented as part of the *conference*-dataset evaluation, the original authors compute the optimal thresholds separately, implying that the alignments have been pre-processed by filtering at the given threshold..

$$F_\beta^T(A,R) = \max_x F_\beta(A_x,R) \tag{2.7}$$

The thresholded F-measure F_β^T eliminates the bias introduced through the application of a threshold. Furthermore, it provides an upper boundary on the possible performance of a particular system.[2] Given a set of alignments To inspect the precision and recall at this upper boundary, we define the thresholded precision and thresholded recall as follows:

$$P_\beta^T(A,R) = P(A_x,R), \textit{where } x = \arg\max_y F_\beta(A_y,R) \tag{2.8}$$

$$R_\beta^T(A,R) = R(A_x,R), \textit{where } x = \arg\max_y F_\beta(A_y,R) \tag{2.9}$$

It may be desirable to inspect the relation between precision and recall in more detail, and observe how this changes with respect to the applied threshold. For this purpose, the thresholded F-measure, precision and recall are insufficient as they only measure the quality of an alignment at a single cut-off point.

The comparison in Table 2.1 provided an overview on how the F-measure behaves at different possible thresholds. A similar and established method exists for the detailed analysis of precision and recall, known as a *precision-recall curve*. This plot is created by sorting all retrieved correspondences according to their confidence and simply plotting the development of precision and recall for all possible cut-off points on a curve. An example *precision-recall curve* is displayed in Figure 2.5.

A distinct pattern that is observable in every *precision-recall graph* is the zig-zag shape of the curve. This stems from the fact that, when the cut-off point is lowered, the added correspondence can only result in both precision and recall increasing, in the case it is correct, or the precision decreasing and the recall staying constant in the case it is incorrect.

In order to increase the clarity of a precision-recall graph, it is common to calculate the *interpolated-precision* at certain recall-levels. To formally define the *interpolated-precision*, we must first define the precision of an alignment at a specified recall value. Given an alignment A, a reference R and a specified recall value r, the precision $P(A,R,r)$ at specified recall r is defined as follows:

$$P(A,R,r) = P(A_x,R), wherex = \max\{y|R(A_y,R) = r)\} \tag{2.10}$$

The *interpolated precision* at recall value r is defined as the highest precision found for any recall value $r' \geq r$:

$$P_{int}(A,R,r) = \max_{r' \geq r} P(A,R,r') \tag{2.11}$$

Note that while the measure of precision is not defined for a recall value of 0, since its computation can result a division by zero if A is empty, the interpolated

[2] When aggregating the quality of a series of alignments, it is also possible to compute a single optimal threshold that is applied to all alignments. This method would represent an upper boundary in which a mapping system cannot dynamically configure itself for each matching task, e.g. by applying a static threshold.

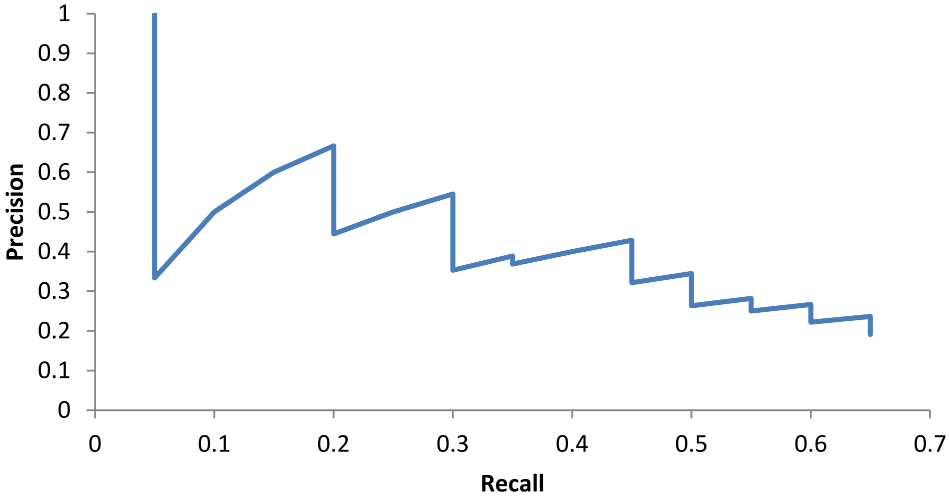


Figure 2.5: Precision-Recall graph

precision is defined for a recall of zero. Furthermore, $P(A, R, r)$ is undefined if $\{y|R(A_y, R) = r)\} = \emptyset$, though $P_{int}(A, R, r)$ is still defined if there exists an r'such that $\{y|R(A_y, R) = r')\} \neq \emptyset$.

Plotting the interpolated precisions has the effect that the zig-zag pattern of the curve is flattened, creating a step-like pattern instead. A common variant of the precision-recall curve is its computation using a set of 11 standard recall levels. These span the interval of $[0, 1]$ with increments of 0.1. Figure 2.6 illustrates a precision-recall curve using interpolated precisions by adding the new curves to the example seen in Figure 2.5.

In a typical information retrieval scenario, precision-recall curves cover the entire spectrum of possible recall values. This is because information retrieval techniques evaluate and rank the entire corpus of documents, meaning that if the cut-off point is set low enough, all relevant document will be retrieved, though usually at a very low precision rate. In ontology mapping, however, a produced alignment is rarely complete with respect to the reference alignment. Thus, the standard precision and interpolated precision are undefined for recall values higher than the highest achievable recall of the given alignment. Therefore, when comparing the precision-recall curves of two different alignments, it can occur that the curves have different lengths.

The measures of precision and recall are well understood and widely adapted. However, an inherent weakness is that they do no account for the 'closeness' of the found correspondences to the reference alignment. The overlap function $R \cap A$ only selects correspondences which are a perfect match with a reference correspondence with respect to the entity pairings, relation type and confidence value. For example, given the reference correspondence $<$ id8, $Car, Car, \equiv, 1.0 >$, one could argue that an incorrect correspondence $< id15, Car, Four\text{-}Wheeled, \equiv, 1.0 >$ is of a better

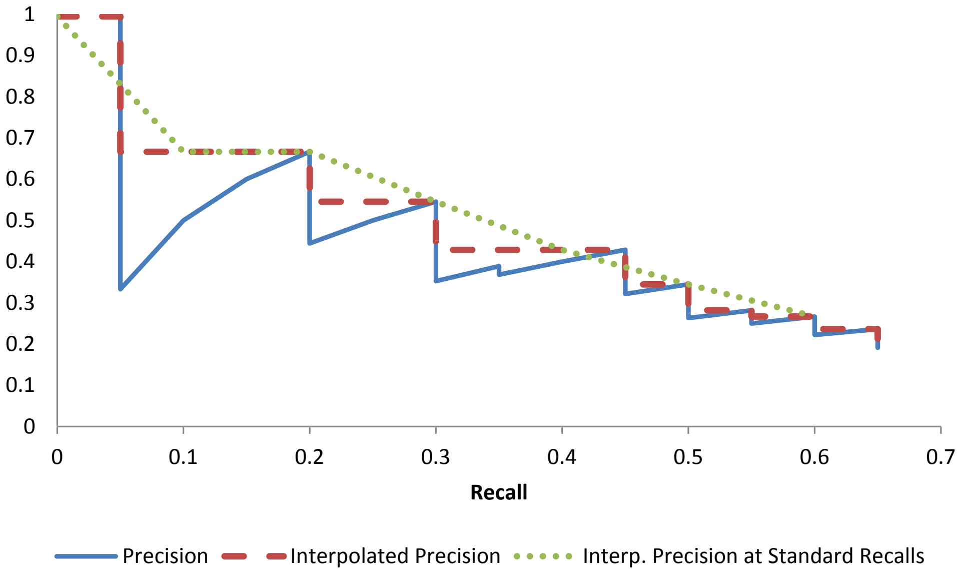


Figure 2.6: Precision-Recall graph. Includes a curve of interpolated precisions for all possible recall values (red) and a curve of interpolated precisions at the standard 11 recall values (green).

quality than an incorrect correspondence $< id16, Car, Bicycle, \sqsubseteq, 1.0 >$, since the concepts of $id15$ can share some common instances, whereas the concepts of $id16$ do not. However, both correspondences are equally filtered out when computing the intersection $R \cap A$, resulting in no observable difference in precision or recall. The closeness of the found correspondences to the reference alignment is of particular importance when considering that in a real-world scenario a domain expert typically inspects and repairs the computed alignment. Here, it would be beneficial to that expert if the repair effort would be as small as possible. To account for this, Ehrig and Euzenat (2005) introduced the measures of *relaxed precision and recall*. These replace the intersection component $R \cap A$ with an *overlap proximity* function w and are defined as follows:

$$P_w(A, R) = \frac{w(A, R)}{|A|} \tag{2.12}$$

$$R_w(A, R) = \frac{w(A, R)}{|R|} \tag{2.13}$$

The aim of the *relaxed precision and recall* is to provide a generalization of precision and recall. Thus, it is possible to select w in such a way that it replicates the results of precision and recall. Ehrig and Euzenat (2005) introduced a straight-forward interpretation of the *overlap proximity*, consisting of the sum of correspondence proximities over a set of selected correspondence pairs. Given an alignment A and a reference alignment R, a pairwise mapping $M(A,R)$ between the correspondences of A and R and a correspondence overlap function $\sigma(a,r)$, the *overlap proximity* between A and R is defined as follows:

$$w(A,R) = \sum_{\langle a,r\rangle \in M(A,R)} \sigma(a,r) \tag{2.14}$$

Given this definition of $w(A,R)$, the problem of computing the *overlap proximity* is de-composed into two sub-problems: (1) computing the correspondence mapping set M and (2) defining a correspondence proximity function.

The set $M(A,R) \subseteq A \times R$ contains a series of correspondence pairings between A and R. In order to preserve the possibility of replicating the standard measures of precision and recall, $M(A,R)$ should be restricted to a subset of $A \times R$ in which any concept may only appear at most once (Ehrig and Euzenat, 2005). The set $M(A,R)$ can be computed using the *Best-Match* policy, which is defined as follows:

$$\begin{gathered} M(A,R) = \operatorname*{arg\,max}_{X \subseteq K} \sum_{\langle a,r\rangle \in X} \sigma(a,r); \textit{where} \\ K = \{C \subseteq A \times R | \forall \{(a,r),(a,r')\} \subseteq C, r = r' \wedge \forall \{(a,r),(a',r)\} \subseteq C, a = a'\} \end{gathered} \tag{2.15}$$

Next, one needs to define the correspondence proximity $\sigma(a,r)$. This function receives as input two correspondences, being $a =< id_a, e_{1,a}, e_{2,a}, t_a, c_a >$ and $r =< id_r, e_{1,r}, e_{2,r}, t_r, c_r >$. A pair of correspondences can differ with respect to the mapped entities e_1 and e_2, identified relation type t or confidence value c. For each of these differences a domain expert would have to perform a repair action in order to fix the malformed correspondence. Thus, $\sigma(a,r)$ would need to take each of these differences into account. This can be done by defining $\sigma(a,r)$ as a combination of three proximity functions $\sigma_{pair}(< e_{1,a}, e_{1,r} >, < e_{2,a}, e_{2,r} >)$, $\sigma_{rel}(t_a,t_r)$ and $\sigma_{conf}(c_a,c_r)$. The correspondence proximity $\sigma(a,r)$ can then be defined as follows:

$$\begin{gathered} \sigma(< id_a, e_{1,a}, e_{2,a}, t_a, c_a >, < id_r, e_{1,r}, e_{2,r}, t_r, c_r >) = \\ \sigma_{pair}(< e_{1,a}, e_{1,r} >, < e_{2,a}, e_{2,r} >) \times \sigma_{rel}(t_a,t_r) \times \sigma_{conf}(c_a,c_r) \end{gathered} \tag{2.16}$$

The combination of σ_{pair}, σ_{rel} and σ_{conf} determines how $M(A,R)$ is selected and ultimately the result of $w(A,R)$. To show that $P_w(A,R)$ and $R_w(A,R)$ are indeed generalizations, we will provide the three proximity functions which can be used to replicate the standard measures of precision and recall. These are collectively referred to as the *equality proximity* and are computed as follows:

$$\sigma_{pair}(< e_{1,a}, e_{1,r} >, < e_{2,a}, e_{2,r} >) = \begin{cases} 1 & \textit{if} < e_{1,a}, e_{2,a} > = < e_{1,r}, e_{2,r} > \\ 0 & \textit{otherwise} \end{cases}$$

$$\sigma_{rel}(t_a, t_r) = \begin{cases} 1 & \textit{if}\ t_a = t_r \\ 0 & \textit{otherwise} \end{cases}$$

$$\sigma_{conf}(c_a, c_r) = \begin{cases} 1 & \textit{if}\ c_a = c_r \\ 0 & \textit{otherwise} \end{cases}$$

There exist numerous ways in which σ_{pair}, σ_{rel} and σ_{conf} can be defined to measure the correction effort. Typically, σ_{pair} returns a non-negative value if the retrieved entities of a are a specialization of generalization of the entities in r. While a detailed survey of all variations of σ_{pair}, σ_{rel} and σ_{conf} is beyond the scope of this work, we suggest the reader consult the work by Ehrig and Euzenat (2005) for example definitions of σ_{pair}, σ_{rel} and σ_{conf}.

2.3 Alignment Evaluation with Partial Alignments

As stated in section 1.4, the main focus of the presented research is the mapping of ontologies while exploiting available background knowledge. For some types of background knowledge, such as lexical resources, ontological resources or parameter sets, the valuation techniques introduced in section 2.2 can be still be applied. However, the evaluation of alignments that have been generated while exploiting a given partial alignment poses a unique challenge. In a typical mapping scenario involving a partial alignment, it is assumed that the correspondences within the partial alignment are correct. Therefore, these correspondences will also be included in the output alignment, since the goal is to create a single complete alignment. Computing the measures of precision, recall and F-measure would thus create a positive bias, since the correspondences of the partial alignment will contribute to an increase in both precision and recall. This bias could obfuscate the true performance of the given system.

As an example, let us assume that we are given the alignment A and reference R from the example in section 2.2. Furthermore, let us assume that we are given the following partial alignment PA which was exploited during the mapping process:

$$PA = \left\{ \begin{array}{r} < \text{id1}, \textit{Vehicle}, \textit{Conveyance}, \equiv, 1.0 > \\ < \text{id2}, \textit{id}, \textit{registration\_number}, \sqsupseteq, 1.0 > \\ < \text{id5}, \textit{licensePlate}, \textit{issued\_plate}, \equiv, 1.0 > \\ < \text{id6}, \textit{Insur.}, \textit{Insurer}, \equiv, 1.0 > \\ < \text{id8}, \textit{Car}, \textit{Car}, \equiv, 1.0 > \end{array} \right\}$$

The correspondences of any given partial alignment are commonly also referred to as *anchors*. The dynamics of the new evaluation problem are visualized in Figure 2.7:

Figure 2.7 depicts all example correspondences according to their association with A, R or PA. Note that PA is a subset of A, since it is assumed that all

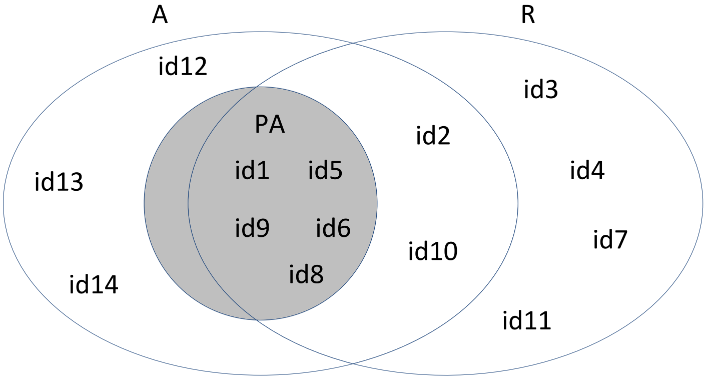


Figure 2.7: Visualization of the dynamics between output, reference and partial alignments of the example.

correspondences of PA will be included in A. However, while all correspondences of PA are also part of R in this example, we did not depict PA as being a subset of R. This stems from the fact that, while it is reasonable to assume that all correspondences of PA are correct, this might not be the case in practise. Chapter 6 will deal with this particular situation in more detail.

Computing the precision and recall of A yields values of 0.7 and 0.6363 respectively. These measurements however are deceiving, as five of the correct correspondences of A were provided in PA and therefore not generated by the tested mapping system. In order to measure the true quality of the correspondences contributed by a given system, one must take the correspondences of PA into account. Ideally, the measurement should reflect the quality of the correspondences in A which are not part of PA. It is possible to adapt the measures of precision and recall such that these take a given partial alignment into account. We refer to this variation of precision and recall as the *adapted precision* $P^*(A, R)$ and *adapted recall* $R^*(A, R)$ (Caraciolo *et al.*, 2008; Schadd and Roos, 2013)[3].

Given an alignment A, a reference alignment R and partial alignment PA, the adapted precision $P^*(A, R)$ with respect to PA can be calculated as follows:

$$P^*(A, R, PA) = \frac{| A \cap R \cap \overline{PA} |}{| A \cap \overline{PA} |} \tag{2.17}$$

The adapted recall $R^*(A, R)$ with respect to PA can be calculated as follows:

$$R^*(A, R, PA) = \frac{| A \cap R \cap \overline{PA} |}{| R \cap \overline{PA} |} \tag{2.18}$$

[3] P^* and R^* are only informally introduced in (Caraciolo *et al.*, 2008) by textually describing the adaptations to P and R and referring to the new measures also as *precision* and *recall*.

Using the measures of adapted precision and recall we can now express the quality of the correspondences that were actually contributed by the mapping system. For our example, this would result in the following measurements:

$$P^*(A,R,PA) = \frac{\mid A \cap R \cap \overline{PA} \mid}{\mid A \cap \overline{PA} \mid} = \frac{|\{\mathit{id2,id10}\}|}{|\{\mathit{id2,id10,id12,id13,id14}\}|} = \frac{2}{5} = 0.4$$

$$R^*(A,R,PA) = \frac{\mid A \cap R \cap \overline{PA} \mid}{\mid R \cap \overline{PA} \mid} = \frac{|\{\mathit{id2,id10}\}|}{|\{\mathit{id2,id3,id4,id7,id10,id11}\}|} = \frac{2}{6} = 0.333$$

Taking the supplied partial alignment into account for our example by calculating the adapted precision and recall, being 0.4 and 0.333 respectively, reveals that the quality of the identified correspondences is not as high as the standard measures of precision and recall, 0.7 and 0.6363 respectively, implied.

Using the measures of adapted precision and recall, we can now define the measure of adapted F-measure, allowing one to express the quality of an alignment using a single value, while accounting for the context of a supplied partial alignment:

$$F^*_\beta(A,R,PA) = (1+\beta^2) \times \frac{P^*(A,R,PA) \times R^*(A,R,PA)}{(\beta^2 \times P^*(A,R,PA)) + R^*(A,R,PA)} \tag{2.19}$$

2.4 Ontology Alignment Evaluation Initiative

The need for techniques which can tackle the ontology mapping problem has been recognized in the scientific community. To stimulate research in this area and to compare existing approaches, the Ontology Alignment Evaluation Initiative[4] (OAEI) (Euzenat *et al.*, 2011b) was founded. This organization hosts yearly competitions in order to evaluate current state-of-the-art system using a series of datasets. Before being known as OAEI, the contest was initially held twice in 2004, first at Information Interpretation and Integration Conference (I3CON) at the NIST Performance Metrics for Intelligent Systems workshop (PerMIS) (Euzenat *et al.*, 2011b), and second at the International Semantic Web Conference (ISWC) during the Evaluation of Ontology-based Tools (EON) workshop (Euzenat, 2004b). In 2005 the contest was name 'Ontology Alignment Evaluation Initiative' for the first time and held at the workshop on Integrating Ontologies during the International Conference on Knowledge Capture (K-Cap) (Euzenat *et al.*, 2005). All subsequent workshops were held at the Ontology Matching workshop, which is collocated with the International Semantic Web Conference (Grau *et al.*, 2013).

During its existence, the OAEI contest has grown steadily. The initial evaluation fielded only 7 participants (Euzenat *et al.*, 2005), while the most recent edition saw 23 systems participating (Grau *et al.*, 2013). A valuable addition to the evaluation campaign was the introduction of the SEALS platform (Trojahn *et al.*, 2010). This

[4] http://oaei.ontologymatching.org/

software platform allows mapping system creators to wrap and upload their tools, such that these can automatically be evaluated and compared to other tools. Furthermore, it promotes the accessibility of the tools for other researchers and supports the validity of the results, since the evaluations are performed by a neutral party and can easily be replicated.

The OAEI competition is run using a series of datasets, where each dataset tests a particular aspect of the ontology mapping problem (e.g. lack of data, ontology size). These datasets are typically created as a response to the current challenges facing the field (Shvaiko and Euzenat, 2008; Shvaiko and Euzenat, 2013), with the intention that each dataset stimulates research in the problem area that this dataset represents. The initial competition was run on only two datasets, while the most recent competition was run using 10 different datasets.

We will provide a brief overview of these datasets in the following subsection.

2.4.1 Datasets

Benchmark

The *benchmark* dataset is one of the oldest datasets used by the OAEI competition. It is a synthetic dataset which consists of matching tasks where one given ontology has to be matched to many different systematic variations of itself. These variations entail the alterations or removal of all possible combinations of meta-information of one ontology. Examples of this are the distortion, translation or removal of concept labels, translation or removal of comments, removal of properties, removal of instances and the flattening or expansion of the concept hierarchy. Hence, this dataset tests the robustness of an approach when faced with a lack of exploitable meta-information.

During the years, this dataset has constantly evolved. The base ontology has been changed numerous times and different types and combinations of alterations were introduced. Another notable change for more recent competitions has been the expansion of this dataset using multiple base ontologies (Aguirre *et al.*, 2012; Grau *et al.*, 2013). These ontologies vary in size and facilitate the observation of the scalability of the tested matching systems.

Conference

The *conference* dataset consists of 16 ontologies modelling the domain of organizing scientific conferences. These ontologies were developed within the *OntoFarm* project. What is different is that all ontologies in this dataset originate from real-world systems, facilitating an estimate on how well a mapping system might perform in a real-world application. The ontologies are quite heterogeneous in terms of structure and naming conventions, providing a challenging environment for the evaluation of mapping systems.

Anatomy

The *anatomy* dataset consists of a single mapping task between two biomedical ontologies, one describing the anatomy of an adult mouse and one being a part of the NCI thesaurus, with this part describing human anatomy. This dataset is noteworthy for several of reasons. One is the size of the given ontologies. Whereas the ontologies of the *conference* dataset contained at most 100 concepts, the ontologies of the *anatomy* dataset are significantly larger, 2744 classes for the mouse ontology and 3304 classes for the human ontology. This dataset thus presents a complexity problem, where a mapping system must provide an alignment within an acceptable time. For the OAEI competition in particular, systems are given one day of computational time to generate an alignment (Grau *et al.*, 2013). Whilst different OAEI contests have offered a variation of this dataset where a partial alignment is provided for the mapping task, unfortunately it has not been executed in recent years due to lack of participants (Aguirre *et al.*, 2012; Grau *et al.*, 2013).

Another challenging aspect is the use of domain specific terminology for the concept labels and descriptions. Hence, there is little natural language present in the concept descriptions, making the application of natural-language processing techniques difficult. Approaches which use thesauri or external ontologies also struggle with this dataset, as external ontologies or thesauri are typically limited to general language concepts and are thus unlikely to contain the concepts which are modelled in the given ontologies. Another differentiator to typical ontologies is the use of specific annotations and roles, e.g. the widespread use of the *partOf* relation.

Library

After being introduced in 2007, the *library* dataset presents a mapping task in which two large thesauri have to be mapped using the SKOS (Miles *et al.*, 2005) mapping vocabulary. In its original version (Isaac *et al.*, 2009), the *library* dataset consisted of two thesauri used by the National Library of the Netherlands (KB) for the indexation of two of its collections. The KB uses the *GTT thesauri* for the indexation of its Scientific Collection, while relying on the *Brinkman thesaurus* for the indexation of its Deposit Collection, containing all Dutch printed publications. The two thesauri contain approximately 35.000 and 5.000 concept descriptions. While both thesauri have a similar domain coverage, they differ greatly with respect to their granularity.

The 2009 variant of this dataset saw the same methodology, however the original ontologies were replaced with the *Library of Congress Subject Headings list (LCSH)*, the *French National Library heading list (RAMEAU)* and the *German National Library heading list (SWD)* (Euzenat *et al.*, 2009a). Here, an additional difficulty is the multi-lingual aspect of the mapping problems.

Whilst not being run in 2010 and 2011, the *library* dataset returned in the 2012 OAEI competition (Ritze and Eckert, 2012). This edition no longer features the multi-lingual aspect of the 2009 edition, with multi-lingual heterogeneity now being tested separately in the *Multi-Farm* dataset. Instead, the 2012 version consists of the *STW* Thesaurus for Economics and *The Thesaurus for Social Sciences (TheSoz).*

MultiFarm

Introduced in 2012, the *MultiFarm* dataset is specifically designed to test a mapping system's capability to match ontologies that are formulated using a different natural language (Meilicke *et al.*, 2012). This dataset has been created by taking a subset of the *conference* dataset and translating the ontologies from English to a series of other languages, being Chinese, Czech, Dutch, French, German, Portuguese, Russian and Spanish. The dataset contains problems of all possible language combinations, and for every combination there exists a problem of mapping two originally different ontologies and a problem of mapping the same ontology that has been translated into different languages. By comparing the results of these two variants one can observe to what extent a system exploits non-linguistic features of the ontology for its results, as the alignment between the same ontology translated into different languages is likely to be of a much higher quality.

Large Biomedical Ontologies

This dataset consists of mapping tasks where large-scale semantically rich biomedical ontologies need to be aligned. Three ontologies are provided, namely the *Foundational Model of Anatomy (FMA)*, the *SNOMED CT* ontology and the *National Cancer Institute Thesaurus (NCI)* consisting of 78.989, 306.591 and 66.724 classes respectively. To create the reference alignments the *UMLS Metathesaurus* (Lindberg, Humphreys, and McCray, 1993) is exploited. Given the large scale of the different matching problems, this dataset is very suitable as a stress test for the efficiency and scalability of a particular mapping system.

Instance Matching

Introduced in 2009 (Euzenat *et al.*, 2009b), the *Instance Matching* dataset saw a rapid evolution during its existence. The main goal of this dataset is to measure instance matching techniques. Here the primary task is not the identification of correspondences between ontology concepts, but instead the identification of instances which are present in both ontologies but actually model the same real world entity. In its initial variation, the dataset consisted of three separate benchmark tests. Two of these were set up using real-world data collections, using the *eprints*, *Rexa* and *SWETO-DBLP* datasets to form the first benchmark and *TAP*, *SWEETO-testbed* and *DBPedia 3.2* to form the second benchmark. The third benchmark is a series of synthetic tests where one collection, namely the *OKKAM* dataset, has to be matched to different variations of itself.

The 2010 edition saw two main modalities (Euzenat *et al.*, 2010), being a data-interlinking track and a OWL data track. In 2011, the dataset was altered such that it offered one data-interlinking track and one synthetic benchmark (Euzenat *et al.*, 2011a). The data-interlinking track consisted of rebuilding links among the *New York Times* dataset itself and identifying shared instances between the *New York Times* dataset and the external resources *DBPedia*, *Geonames* and *Freebase*. The synthetic track is created by introducing systematic alterations to the *Freebase* data. The 2011 edition of the *instance matching* dataset has also been use for the 2012 and 2013 campaign. However, to address the problem of finding similar instances, instead of identical instances, the 2014 edition[5] of this dataset was set-up to include an identity recognition track and a similarity recognition track.

Interactive Matching

The *interactive matching* evaluation was introduced in 2013 in order to address the need for the evaluation of semi-automatic approaches (Grau *et al.*, 2013). This stems from the intuition that a user can be a valuable asset for improving the quality of the generated alignments. The set-up for this dataset differs from a typical partial alignment mapping problem. Instead of being given a series of correspondences, the given system may consult a user regarding the correctness of a correspondence. To perform this evaluation automatically, the given user is simulated through the use of an Oracle class, which can check suggested correspondences by inspecting the reference alignment. The ontologies used for this dataset stem from the *conference* dataset. By comparing a system's results of the *interactive matching* dataset with the results of the *conference* dataset, one can observe how much the interactive component of the system actually contributes to the alignment quality.

Query Answering

This newest addition to the OAEI campaign aims to present an alternative to the typical reference-based evaluation of alignments (Solimando, Jiménez-Ruiz, and Pinkel, 2014). Instead, the goal is to evaluate mapping systems in an ontology-based data access scenario. This set-up is comparable to an information integration scenario, as introduced in section 1.2.2. In essence, the systems are required to compute an alignment between a query ontology and a local ontology. A series of queries are translated using the alignment produced by the system, such that an answer set is generated for each query. These answer sets can be evaluated by comparing them to the desired answer sets by computing the measures of precision, recall and F-measure.

[5] results paper publishing in progress

Chapter 3

Mapping Techniques

In the previous chapter we have formally introduced the problem of ontology mapping and evaluation techniques. In this chapter, we will introduce the basic techniques that are commonly used to create such mapping functions. In section 3.1 we will introduce basic system architectures, a categorization of similarity techniques, and a brief overview of similarity aggregation and correspondence extraction techniques. In section 3.2 we will provide a survey of the contemporary state-of-the-art mapping systems.

3.1 Basic Techniques

The core task of an ontology mapping system is the identification of correspondences that exist between the concepts of two given ontologies. In order to achieve this, the system must be able to determine the likelihood of two concepts being used to encode the same or similar information. This task is usually achieved through the usage of *similarity* measures. Using the results of the similarity measures, the system must determine the overall similarity between the given concepts and decide which possible correspondences to include in the output alignment. Thus, the core tasks which are performed during ontology mapping can be summarized as follows:

Similarity Computation Computation of similarities between the ontology concepts. A similarity measurement exploits the encoded meta-information of concept definitions in order to produce its measurements. Various types of meta-information can be used for this purpose, such as concept labels, descriptions, comments, related concepts or provided instantiations of that concept. While a similarity measurement is typically correlated with the statistical likelihood of two given concepts denoting the same entity, it is not a statistical estimate, i.e. the computed measurements over the entire problem space are not normalized and do not take previously observed measurements into account.

Similarity Combination It is common for mapping systems to employ multiple similarity measurements. The reason for this is that similarity metrics typically exploit a limited range of available meta-information and determine the similarity between concepts using a specific intuition. If one of these two aspects fails then the similarity will be very unlikely to reflect appropriate similarity measurements. An example of the aspect of information availability failing would be a mapping system employing a comment-based similarity to process an ontology which does not contain concept comments. An example of the similarity intuition aspect failing is the use of a string-similarity when the concept names are expressed using synonyms. The usage of multiple similarity metrics means that for each combination of concepts there will be multiple measurements of their similarity. At this step, it is necessary to combine these measurements using a specific similarity combination technique.

Correspondence Extraction After the similarities between all possible concept combinations have been converted into a single value, it becomes necessary to determine which correspondences will be included in the output alignment. Whether a specific correspondence linking the concepts x and y will be included is determined by not only inspecting its own similarity value, but also by analysing all possible correspondences which link either x or y to alternative concepts. Alternatively, one can analyse correspondences which map the respectively related concepts of x and y, e.g. the possible correspondence between the parent concept of x and the parent concept of y.

Using these core functions, we can now visualize the entire mapping process in more detail as can be seen in Figure 3.1.

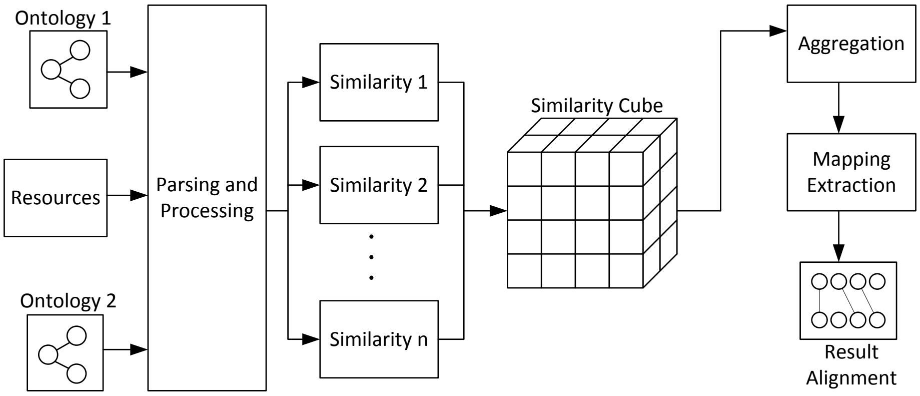


Figure 3.1: Basic architecture of an ontology mapping framework.

In Figure 3.1 we can see the entire ontology mapping process. On the left side we can see the inputs of the mapping problem, being two ontologies and the set of additional resources. The first task to be performed involves the parsing and processing of the input ontologies. Here, the ontologies are parsed into a format that is supported by the system. For example, if the system is designed with OWL-DL ontologies in mind and receives a RDF-schema as one of the input ontologies, then the RDF-schema will be parsed into an OWL-DL ontology during the parsing step. Furthermore, several pre-processing steps can be performed at this step. Examples of these are word-stemming, stop-word filtering, part-of-speech tagging, synonym identification or the creation of semantic annotations. After this step, all configured similarity measures will proceed to calculate all pairwise similarities between the concepts of both ontologies. When two ontologies consist of x and y concepts respectively, the result of the similarity computation step using n similarity metrics is a $x \times y \times n$ cube of similarity measures. It is possible to apply special partitioning techniques at this stage. Partitioning techniques attempt to reduce the mapping problem into smaller sub-problems such that the similarity cube does not have to be computed completely without impacting the alignment quality (Stuckenschmidt and Klein, 2004), increasing the efficiency of the system. Since only a subset of the $x \times y \times n$ matrix is actually computed, one can compile the results into a sparse $x \times y \times n$ matrix.

Next, the $x \times y \times n$ cube is reduced to a $x \times y$ matrix of similarity measures through the use of an aggregation technique. Examples of these techniques are statistical measures or machine learning techniques. These will be introduced in more detail in Section 3.1.3. In the third core step, the $x \times y$ similarity matrix, is used to determine which pairwise combinations of concepts will be included in the result alignment. Here, the aggregated similarity value of a concept pair is typically also used as the confidence value of the respective correspondence.

The mapping steps starting from the parsing and preprocessing sub-process and ending with the mapping extraction, as seen in Figure 3.1, can be interpreted as the mapping function a system, as defined in Definition 4. The example of Figure 3.1 however only portrays a straight forward approach of such a function. It is of course possible to structure a mapping system differently while still conforming to the definition of an ontology mapping system. We will introduce a selection of prolific system structure alternatives which utilize two ontologies 0_1 and O_2, an input alignment A, a set of parameters p and a set of resources r in order to produce an output alignment A'.

Figure 3.2 visualizes the basic principle behind an *iterative system*. Here, the output alignment A' serves as input for an additional execution of the mapping system. Typically, the system produces a small set of high precision correspondences. These serve as a basis for the discovery of related correspondences which themselves might not exhibit a high pairwise similarity. The iteration process continues until a certain stopping criteria is met. Such a criteria can be the stability of the mapping, i.e. no significant changes to the alignment after an iteration, or a limit on the amount of iterations or maximum runtime, etc.

3.1.1 System Composition

When given a selection of mapping systems, it becomes possible to create a meta-solution to the mapping problem by composing the individual systems into a single system. One way of achieving this is through *sequential composition.* Given two

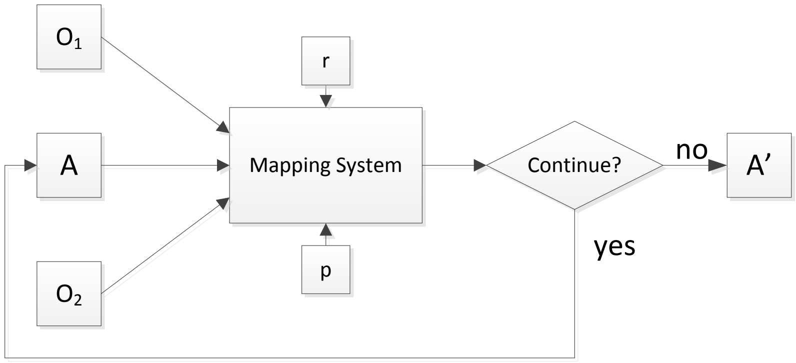


Figure 3.2: An iterative mapping system.

systems, being *Mapping System 1* and *Mapping System 2*, a *sequential composition* first presents the mapping problem between O_1 and O_2 to *Mapping System 1*. This system produces an output alignment A'. *Mapping System 2* then utilizes A' as input alignment for the task of mapping O_1 with O_2 in order to produce the output alignment A''. This process is illustrated in Figure 3.3.

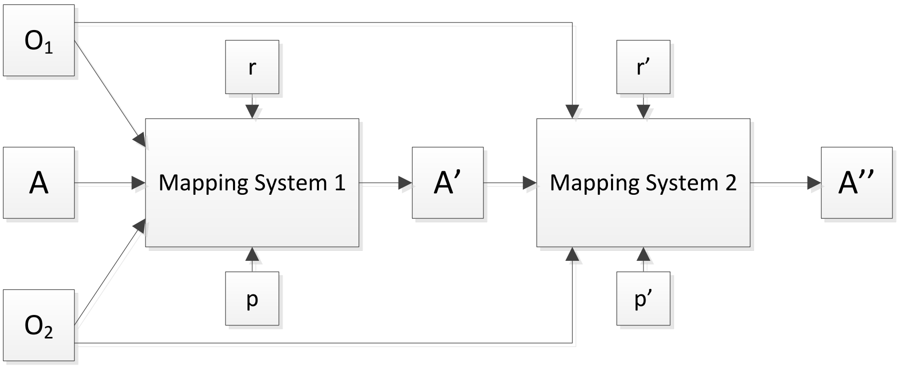


Figure 3.3: A sequential composition of mapping systems.

Mapping approaches which exploit partial alignments are commonly deployed in sequential compositions, specifically as part of *Mapping System 2*. Here, *Mapping System 1* typically serves as a means to generate a partial alignment for the particular approach in the case that no partial alignment is provided. If such an alignment does exist, *Mapping System 1* can either opt to forward that alignment to *Mapping System 2* or attempt to enhance it, for instance through the discovery of additional correspondences or the verification of the input alignment.

An alternative way of composing mapping systems is through *parallel composition*. Here, the inputs O_1, O_2 and A are forwarded to the given mapping systems *Mapping System 1* and *Mapping System 2*. In this type of composition, the given systems are executed independently of each other, resulting in the output alignments A_1 and A_2. The principle behind a *parallel composition* is illustrated in Figure 3.4.

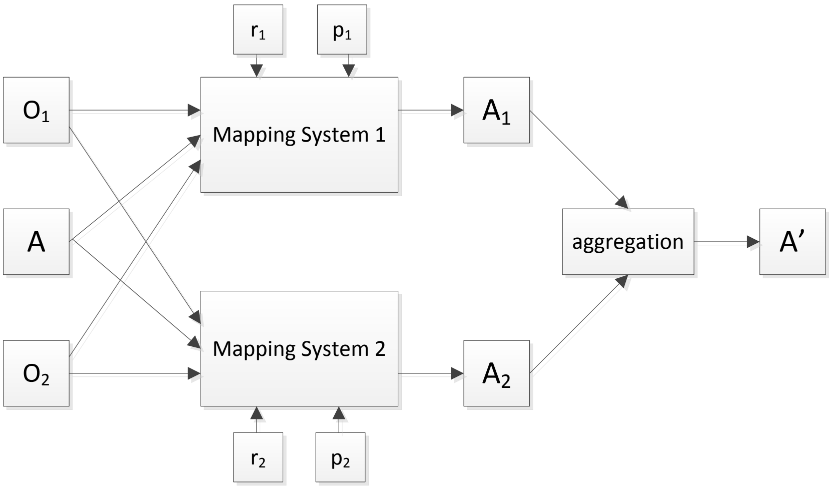


Figure 3.4: A parallel composition of mapping systems.

After the individual systems have been executed, an aggregation technique merges the alignments A_1 and A_2 into a single alignment A', representing the output alignment of a *parallel composition*. There exist a variety of techniques which can be applied in this situation. For instance, a decision system can simply choose between A_1 and A_2 based on some given criteria. Alternatively, the aggregation method can opt to create A' by only selecting correspondences which appear in both A_1 and A_2. A selection can also take place based on the provided confidence values of the correspondences. Threshold-based methods are a good example of such an aggregation, where correspondences are only added to A' if their confidence values satisfy the given threshold. Alternatively, one can create a bipartite matching problem using the supplied alignments. In a bipartite matching problem one is given a graph with two groups of vertices A and B and a set of weighted edges E which only vertices of A with vertices of B. The main task is then to find a subset of E which satisfies a certain criteria. To merge the alignments A_1 and A_2 one can define E such that it only contains an edge e linking the concepts c_1 and c_2 if either A_1 or A_2 contain a correspondence between c_1 and c_2. Executing maximum-weight matching or stable-marriage matching, as detailed in sub-sections 3.1.4 and 3.1.4 can then be used to determine which correspondences are included in the final alignment.

When performing a confidence-based aggregation, it is likely a better option to perform the aggregation step on the produced similarity matrices of *Mapping System 1* and *Mapping System 2* instead on their output alignments A_1 and A_2. The methods which can then be applied are similar to the aggregation techniques discussed in Section 3.1.3.

3.1.2 Similarity Metrics

An essential component of an ontology mapping system is the set of applied similarity metrics. These measure how similar two given ontology entities e_1 and e_2 are with respect to their provided meta-data. We define a similarity metric as follows:

Definition 6 (**Similarity**). *A similarity metric is any function $f(e_1, e_2) \in \mathbb{R}$ which, given a set of entities E, maps two entities e_1 and e_2 to a real number and satisfies the following properties:*

$$\begin{array}{lr}
\forall x \in E, \forall y \in E, f(x,y) \geq 0 & \textit{(positiveness)} \\
\forall x \in E, \forall y \in E, f(x,y) \leq 1 & \textit{(normalized)} \\
\forall x \in E, \forall y, z \in E, f(x,x) \geq f(y,z) & \textit{(maximality)} \\
\forall x \in E, f(x,x) = 1 & \textit{(self-maximality)} \\
\forall x, y \in E, f(x,y) = f(y,x) & \textit{(symmetry)}
\end{array}$$

Similarity metrics are typically executed on concepts originating from different ontologies. However, the origins of the inputs are not a requirement for a function to satisfy the criteria of a similarity metric.

Some techniques measure the dissimilarity between entities and convert resulting value into a similarity value. A dissimilarity is defined as follows:

Definition 7 (**Dissimilarity**). *A dissimilarity metric is any function $f(e_1, e_2) \in \mathbb{R}$ which maps two entities e_1 and e_2 to a real number and satisfies the following properties:*

$$\begin{array}{lr}
\forall x \in E, \forall y \in E, f(x,y) \geq 0 & \textit{(positiveness)} \\
\forall x \in E, \forall y \in E, f(x,y) \leq 1 & \textit{(normalized)} \\
\forall x \in E, f(x,x) = 0 & \textit{(self-minimality)} \\
\forall x, y \in E, f(x,y) = f(y,x) & \textit{(symmetry)}
\end{array}$$

A more constrained notion of dissimilarity can be found in a *distance* measure or in an *ultrametric*, which are defined as (Euzenat and Shvaiko, 2007):

Definition 8 (**Distance**). *A distance metric is any dissimilarity function $f(e_1, e_2) \in \mathbb{R}$ which additionally satisfies the following properties:*

$$\begin{array}{lr}
\forall x, y \in E, f(x,y) = 0 \textit{ iff } x = y & \textit{(definiteness)} \\
\forall x, y, z \in E, f(x,y) + f(y,z) \geq f(x,z) & \textit{(triangular inequality)}
\end{array}$$

Definition 9 (**Ultrametric**). *An ultrametric is any distance function $f(e_1, e_2) \in \mathbb{R}$ which satisfies the following property:*

$$\forall x, y, z \in E, f(x,y) \leq \max(f(x,z), f(y,z)) \quad \textit{(ultrametric inequality)}$$

Some authors define the measures of similarity and dissimilarity without the normalization and self-maximality properties (Euzenat and Shvaiko, 2007), and instead define a normalized version of either metric separately. While the lack of normalization and self-maximality might not cause issues in other scientific domains, in the domain of ontology mapping the lack of these properties can be the cause of issues when combining several metrics into a single mapping system. One would need to account for the output interval of each metric separately such that these can be aggregated into a single appropriate confidence value. Additionally, the lack of self-maximality makes it difficult to express whether the system considers two entities to be identical. It would thus be possible to define multiple similarity metrics which assign different values to identical concept pairings, forcing the system to know which values are used to express which concept pairs are considered to be identical for each metric. Certain matching techniques rely on this knowledge in order generate an input-set of correspondences with the highest degree of confidence. The property of self-maximality is important, for instance, for systems which generate anchors on-the-fly and only wish to exploit correspondences which are considered equal by the given similarity metric. It is for these reasons that most contemporary mapping systems do enforce the normalization and self-maximality properties, justifying their inclusion in Definition 6.

A similarity function as defined in Definition 6 does bear some resemblance to a probability mass function. One might even be inclined to describe a similarity as the probability of two concepts being equal. However, a similarity function is not a probability mass function. While a concept similarity function is likely positively correlated with the theoretical probability of concept equality, it does not model the chance of an event occurring. An intrinsic property of a probability mass function is the normalization over the entire input set. Specifically, a function $p_X(x \in X, y \in X) \to [0,1]$, with X being the set of all possible discrete events, is only a probability mass function if the equality $\sum_{x,y \in X} p(x,y) = 1$ holds. Recall that f is subject to the self-maximality property, meaning that $f(x,y) = 1$ for all x and y which describe a pairwise combination of identical concepts. If X contains n concepts, with $n \geq 2$, then there exist at least n combinations resulting in a value of 1, due to concepts being compared to themselves. Therefore, we know that $\sum_{x,y \in X} f(x,y) \geq n > 1$, contradicting the requirement that the sum of all possible inputs be 1 if f were to be a probability mass function.

There exists a variety of techniques that can be applied as a similarity function. These exploit a variety of algorithms and underlying principles in order to derive their similarity values. One can classify similarity metrics according to certain distinct properties of each metric (Rahm and Bernstein, 2001; Shvaiko and Euzenat, 2005). These classification properties are referred to as *matching dimensions* by Shvaiko and Euzenat (2005). *Matching dimensions* examine the metrics with respect to their expected input format, e.g. XML, RDF, OWL, relational-models, attributes of the formal algorithms and provided output dimensions. While there exist several ways of selecting *matching dimensions* in order to classify similarity metrics, we will illustrate a classification of techniques using dimensions which are relevant to the performed research. The classification, illustrated in Figure 3.5, is based on the works by Shvaiko and Euzenat (2005), particularly the *granularity/input* interpretation of the work. However, the order of the mapping dimensions has been altered in order to cluster mapping techniques which utilize external resources. This grouping emphasizes the breadth of existing resource-based techniques. Additionally, the classification has been updated such that it includes categories which more accurately reflect related work and recent developments.

Our classification utilizes two mapping dimensions, which separate the individual techniques according to the following criteria:

Syntactic/External/Semantic This mapping dimension was introduced in the work by Shvaiko and Euzenat (2005) and distinguishes between interpretations of the input. Syntactic techniques interpret their input as a construct and compute their outputs according to a clearly defined algorithm. External techniques utilize resources in addition to the two given ontologies. Concepts are evaluated by using the provided resource as context. External techniques may exploit resources such as domain specific ontologies or lexicons, partial alignments or human inputs. Semantic techniques utilize the semantic interpretations of the input. Typically, logic-based reasoners are used to infer correspondences, ensuring that the alignment is coherent and complete.

Element-level/Structural-level Element-level techniques function by comparing concepts of different ontologies in isolation, thus omitting information or instances of semantically related concepts. Structure-level techniques do also utilize information or instances of related concepts, thus analysing how concepts appear as part of a structure. This matching dimension was first introduced by Rahm and Bernstein (2001). For instance-based techniques, this dimension was first utilized in the work by Kang and Naughton (2003).

We will provide a brief description of the illustrated techniques from Figure 3.5 in the remainder of this subsection. For more detailed discussions of each technique and demonstrations of particular algorithms, we suggest that the reader consult the works by Ehrig (2006) or Euzenat and Shvaiko (2007).

String-based

String-based techniques evaluate concept pairings by comparing the names and/or descriptions of concepts. These interpret the input as a series of characters. The data these techniques exploit is typically extracted from the `rdf:label` or `rdf:description` properties of the concept descriptions. The main intuition behind string-based techniques is that concepts are more likely to denote the same entity if their names are more similar. String-based techniques typically evaluate metrics such as common prefix or suffix length, edit-distances and n-gram overlap (Cohen, Ravikumar, and Fienberg, 2003).

Language-based

Language-based techniques treat the provided input as an occurrence of some natural language. Thus, language-based similarities apply techniques originating from the

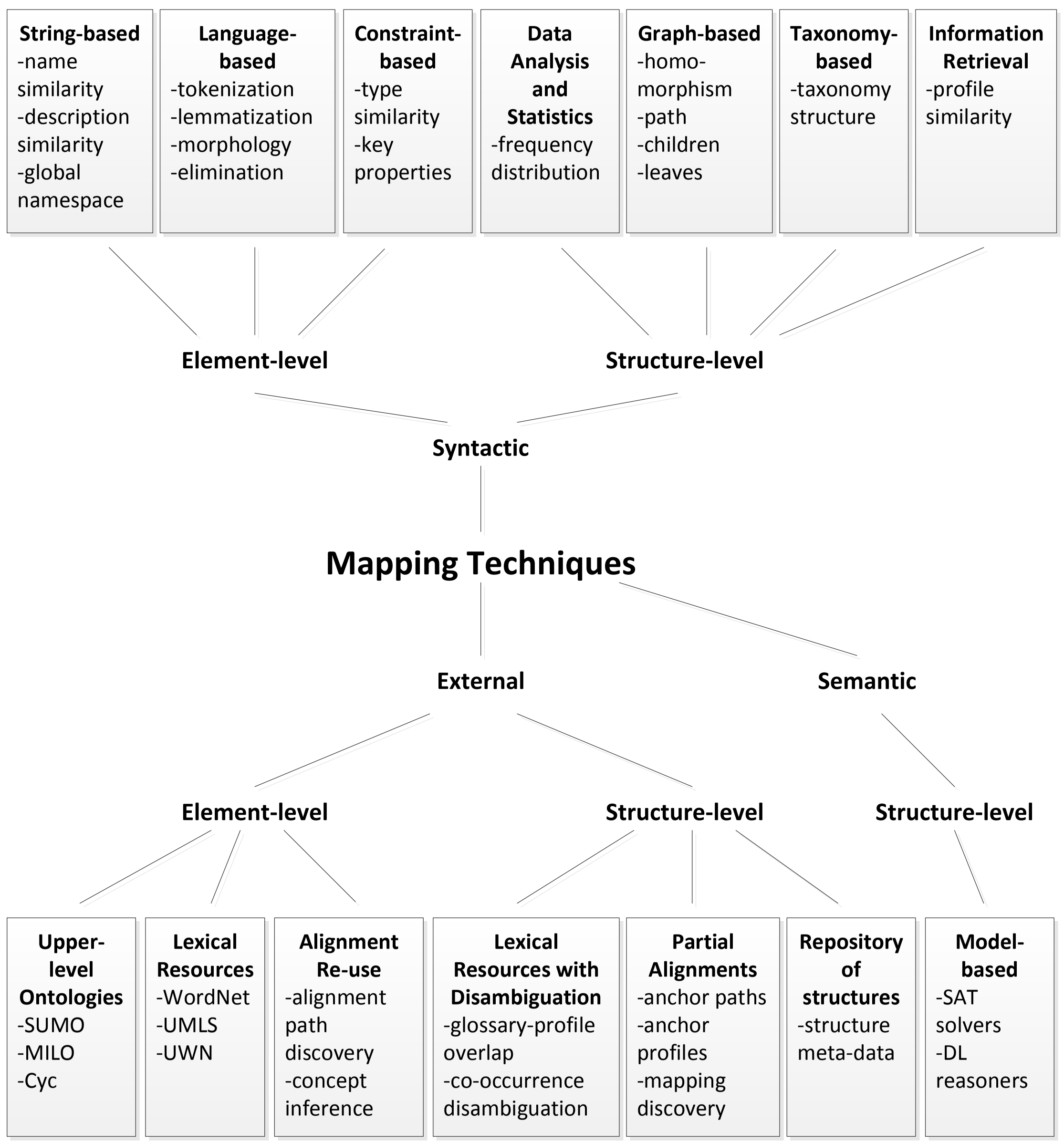


Figure 3.5: Classification of concept mapping approaches. The classification is hierarchically structured, with the top level distinguishing the input interpretation and the bottom level featuring input scope.

field of Natural Language Processing (NLP) in order to extract and process meaningful terms from the concept names or descriptions. Examples of NLP techniques are tokenization, lemmatization, or stop-word removal. The goal of a tokenization technique is to split up compound words into meaningful fragments. For example, a tokenization technique would split up the term *SportsCar* into the token set {*Sports, Car*}, making it easier to establish that the concept of *SportsCar* is in some way associated with the concept *Car*. The process of lemmatization allows for the linguistic reduction of a word into its base form. For example, the two corresponding properties *have* and *has* are unlikely to produce a high result when used as input to a string similarity. By applying lemmatization, one can reformulate the name *has* into its linguistic base form *have*, thus yielding a perfect match. The aim of stop-word removal is to exclude terms which themselves do not carry any meaning from future evaluations, words such as *'a'*, *'the'* or *'that'*. For example, a comparison between two properties *written* and *hasWritten* would yield a more appropriate result after the stop-word *'has'* is removed from the name *hasWritten*.

Constraint-based

Constraint-based techniques compare the internal structure of ontology concepts. The core intuition is that concepts which model the same information are more likely to be similarly structured. Constraint-based techniques analyse concepts with respect to the modelled data-type, e.g. `string`, `int` or `double`, and cardinality of each property. The cardinality of a property describes a set of restrictions for that property, modelling the minimum and maximum amount of times the property may be used for any concept instance. Examples of cardinalities are 1..1, 0..1 and 0..*. The cardinality of 1..1 implies that a property must be present exactly once for each instance. Properties used as identifier are typically associated with this cardinality. A property which may be used at most once is associated with the cardinality 0..1, while a property which may be omitted or occur an arbitrarily large number of times will be associated with the cardinality 0..*. Datatypes and cardinalities are typically compared using a compatibility table (Lee *et al.*, 2002).

Data Analysis and Statistics

Techniques which fall under this category exploit the set of provided instances of the given ontologies. The intuition behind such techniques is that two identical concepts are likely to model the same or at least similar data. Data analysis techniques assume that there is a subset of instances which is modelled in both ontologies. The set of shared instances over the entire ontologies can either be established by exploiting unique key properties, e.g. registration numbers, or assembled using instances mapping techniques (Elfeky, Verykios, and Elmagarmid, 2002; Wang *et al.*, 2008). Ontology concepts can then be compared by analysing the overlap with respect to their shared instances. This can be done by for example computing the Hamming distance (Hamming, 1950) or Jaccard similarity (Jaccard, 1901) between their respective instance sets.

In cases where a shared instance set cannot be computed, it is possible to apply statistical techniques in order to approximate the similarity between instance sets.

The aim is to extract statistical features from the ontology properties using the given instance data, for example the minimum value, maximum value, mean, median, variance, existence of null values, number of decimals or number of segments. The intuition here is that, given a statistically representative set of samples, these characteristics should be more similar, if not the same, for two classes if these denote the same entity.

Graph-based

Graph-based techniques interpret the two input ontologies as a pair of graphs $G_1 = \{V_1, E_1\}$ and $G_2 = \{V_2, E_2\}$. A graph G is characterized by its set of vertexes V and set of edges E. By parsing the ontologies into a graph structure, the problem of mapping two ontologies can then be reformulated as a *graph homomorphism* problem (Bunke, 2000; Fan *et al.*, 2010). The core problem of *graph homomorphism* is to find a mapping from V_1 to V_2, such that each node of V_1 is mapped to a node of V_2 of the same label and each edge in E_1 is mapped to an edge in E_2.

The common notion of a *graph homomorphism* problem is often too restrictive as it aims to produce a full mapping $V_1 \rightarrow V_2$ and $E_1 \rightarrow E_2$. In real-world applications a perfect match between the input structures might not always be possible (Bunke, 2000). In the domain of ontology mapping, this is also the case as ontologies can be modelled with a different scope or granularity. Therefore, graph-based techniques often interpret the problem at hand as a sub-problem of *graph homomorphism*, referred to as the *Maximum common directed subgraph problem* (Levi, 1973). Here, the aim is to find the maximum sets $F_1 \subseteq E_1$ and $F_2 \subseteq E_2$ and the function pair $f : V_1 \rightarrow V_2$ and $f^{-1} : V_2 \rightarrow V_1$ between two graphs $G_1 = \{V_1, E_1\}$ and $G_2 = \{V_2, E_2\}$. This approach to the given graph matching problem allows for a mapping between graph structures even if these differ with respect to their scope or granularity. However, while in classic applications the aim is to maximize the size of the common subgraph, in ontology mapping the problem is typically maximized over a pairwise similarity function over all concept pairings (Euzenat and Shvaiko, 2007). The similarity function typically compares two concepts c_1 and c_2 by comparing the neighbourhood of c_1 within G_1 with the neighbourhood of c_2 within G_2. The core intuition is that the more similar the neighbourhoods of c_1 and c_2 are, the more likely it is that c_1 and c_2 denote the same concept.

Taxonomy-based

Instead of mapping entire ontologies using their graph structures, one can apply the graph matching techniques merely on the taxonomies of the ontologies (Valtchev and Euzenat, 1997). A taxonomy only considers edges representing `subClassOf` relations, which forms the back-bone of the ontology. When interpreting a class as a set of instances, the `subClassOf` relation essentially links two sets where one set is subsumed by the other. The intuition here is that concepts connected with a `subClassOf` relation are semantically very similar. For two concepts $c_1 \in O_1$ and $c_2 \in O_2$ denoting the same entity one can expect that their taxonomic neighbourhoods are also highly similar.

Information Retrieval

Techniques which fall under this category utilize information retrieval (IR) approaches in to derive concept similarities. Ontology concepts are interpreted as documents and each document is populated with relevant information about its respective concept (Mao, Peng, and Spring, 2007; Qu, Hu, and Cheng, 2006). This information can stem from the concept definition, the definitions of related concepts or provided concept instances. The core intuition here is that concept documents are more likely to be similar if their respective concepts model the same entity. Document similarities are computed using information retrieval techniques (Manning, Raghavan, and Schütze, 2008). Typically this includes the weighting of the document terms, for instance using weighting schemes like *Term Frequency-Document Frequency*(*TF-IDF*) or a weighted model utilizing the origin of each term from their respective ontologies. An example of such a weighted model will be introduced in Section 4.3.3.

Upper-level Ontologies

Upper-level ontologies model an abstract domain of concepts using formal semantics. The concepts of such an upper-level ontology can be used as the basis for the creation of a domain specific ontology. A technique utilizing an upper-ontology as background knowledge would hence derive correspondences which are semantically consistent and complete. While there exist several specifications of upper-level ontologies, e.g. SUMO (Niles and Pease, 2001), MILO (Niles and Terry, 2004), Cyc (Matuszek *et al.*, 2006) or DOLCE (Gangemi *et al.*, 2003), research in techniques which utilize these ontologies has so far been unsuccessful (Noy, 2004; Euzenat and Shvaiko, 2007).

Lexical Resources

Lexical resources provide domain specific background to the given matching task. These resources typically model the given domain in great detail and include domain specific terminology. The information of such a resource can be used to enrich the meta-information of each ontology concept by linking each concept with its appropriate entity from the lexical resource. A resource-based technique can then determine the similarity between two concepts by establishing their distance within the given lexical resource. The core intuition behind this category of techniques is that the closer two concepts are within the resource, the more likely they are to denote the same entity. By converting the computed distances into a similarity, one can essentially derive a measure of *semantic relatedness* or *semantic similarity* between two concepts (Strube and Ponzetto, 2006).

Examples of such lexical resources are WordNet (Miller, 1995), YAGO (Suchanek, Kasneci, and Weikum, 2007), UMLS (Bodenreider, 2004) and FMA (Rosse and Mejino Jr, 2003). WordNet and YAGO are examples of broadly defined resources, modelling a very wide domain. For WordNet and YAGO the modelled domain is the general English language. UMLS and FMA are examples of narrow resources which model a specific domain in great detail, being the biomedical domain in the case of UMLS and FMA. Techniques which utilize these types of resources will be further discussed in Chapter 4.

Alignment Re-use

Alignment re-use techniques utilize complete alignments in order to derive concept correspondences using reasoning techniques (Rahm and Bernstein, 2001). In the most basic form, alignment re-use techniques tackle the problem of mapping O_1 with O_2 by exploiting two existing alignments A_1 and A_2, with A_1 specifying a mapping $O_1 \leftrightarrow O_3$ and A_2 specifying the mapping $O_2 \leftrightarrow O_3$. While ideally A_1, A_2 and O_3 are provided as input to the mapping problem, this might not always be the case. Without the provided necessary resources, a mapping system might consult a mapping infrastructure, as described in subsection 1.3.7, in order to automatically identify an ontology O_3 for which appropriate alignments exist. More advanced techniques exploit alignments spanning a variable path length (Aumueller *et al.*, 2005), e.g. $O_1 \leftrightarrow O' \leftrightarrow O'' \leftrightarrow \cdots \leftrightarrow O_2$, or merge the results of several alignment paths (Sabou *et al.*, 2008).

Lexical Resources with Disambiguation

Similarities of this category share some overlap with the previously described lexical resource-based techniques. They utilize the same type of resources, e.g. WordNet or UMLS, and use the same techniques to derive the similarities between entries of these resources. However, the key difference here is that a disambiguation technique is applied before the execution of the lexical similarity. Such techniques use contextual information of a given concept, e.g. labels, descriptions or information of related concepts, in order to identify the correct sense for that given concept. Because disambiguation techniques typically utilize information of related concepts as context to aid the disambiguation step, lexical similarities that utilize a disambiguation step are classified as structure-level approaches instead of element-level. Disambiguation techniques and their integration into a lexical similarity will be further discussed in Chapter 4.

Partial Alignments

Partial alignment-based techniques utilize incomplete alignments between the two given ontologies O_1 and O_2. Such an alignment can stem from a domain expert being unable to finish the complete alignment, or from a mapping system. Such a system might even be designed for the specific purpose of generating a reliable set of correspondences to be used as partial alignment. The main goal here is to exploit the given partial alignment in order to derive additional correspondences, such that a complete alignment can be created. Specific techniques may for instance use the partial alignment as a starting point for the discovery of new correspondences (Seddiqui and Aono, 2009) or explore concept paths between anchors (Noy and Musen, 2001). This category of techniques will be further discussed in Chapter 5.

Repository of Structures

Previously described *alignment re-use techniques* either utilized partial alignments between the given ontologies O_1 and O_2 or complete alignments between either O_1 or O_2 and other ontologies from a repository. Techniques in this category utilize mappings of ontology fragments. Here, both O_1 and O_2 are decomposed into fragments such that similar fragment pairs are identified. For each fragment pair a repository of fragments is consulted for fragments which are similar to both fragments of the given pair. The core intuition here is that alignments to similar fragments can be exploited such that the mapping problem can be shifted into a fragment pair that is easier to match (Rahm, Do, and Massmann, 2004).

Model-based

Model-based techniques process ontology concept based on their semantic interpretation. These techniques use formal semantic frameworks, e.g. propositional satisfiability (SAT) (Giunchiglia and Shvaiko, 2003) or description logic techniques (Bouquet *et al.*, 2006), in order to derive correspondences. These techniques are typically applied in conjunction with a set of anchors since otherwise their performance is unsatisfactory (Euzenat and Shvaiko, 2007). For example, SAT solvers build matching axioms for each pairwise combination of concepts c_1 and c_2 according to a specified relation $r \in \{\equiv, \sqsubseteq\sqsupseteq, \perp\}$. Each mapping axiom is then verified by assuming the negation of that axiom and deriving whether the negation is satisfiable. A mapping axiom is considered true if its negation is unsatisfiable.

3.1.3 Similarity Aggregation

In the previous subsection we introduced a varied selection of techniques that can serve as the basis of a similarity measure for an ontology mapping system. These can be categorized by their different underlying principles and intuitions. However, a single similarity metric does not always produce a reliable result. For example, consider the concepts *Bus* and *Coach* describing the same entity. A human would immediately conclude that these two concepts should be matched. However, depending on the used similarity metric the result can be vastly different. A lexical similarity, as described in Subsection 3.1.2, would quickly identify that these two words are synonyms and therefore produce a similarity value of 1. On the other hand, a string-based technique would return a very low value since the words *'Bus'* and *'Coach'* have no character overlap. Based on this example one should not conclude that a string-based technique is a sub-standard technique. Given the concept pair *Bus* and *Buss*, with the second concept containing a spelling error, a string-based approach would yield a more accurate result since a lexical similarity would be unable to find a concept labelled *Buss* in its corpus.

To overcome the different types of heterogeneities between ontology concepts, a varied selection of mapping techniques is typically applied. For any given concept pair the given mapping system applies every configured similarity metric. The results of these metrics are combined using an *aggregation technique*. In this subsection we will provide a brief overview on techniques that are commonly utilized for the purpose of similarity aggregation. Note that some aggregation techniques are defined over distance measures instead of similarities (Euzenat and Shvaiko, 2007), though in practise this is barely a hindrance since a similarity metric can easily be converted into a distance measure.

Weighted Product

One way of aggregating similarities is by using a product. The similarities can be weighted using a parameter ω, allowing certain metrics to be emphasized over others. For example, one might find a metric based on instance overlap more trustworthy than a metric comparing the internal structure of concepts. Given a set of n similarity metrics $S = \{sim_1, sim_2, \dots, sim_n\}$, a set of weights $\{\omega_1, \omega_2, \dots, \omega_n\}$ and two concepts x and y, an aggregation function $a(x, y, S)$ can be defined as a weighted product as follows:

$$a(x, y, S) = \prod_{i=1}^{n} sim_i(x, y)^{\omega_i} \tag{3.1}$$

Due to its use of multiplication, this function has the unfortunate effect that its result is 0 if only one of the similarities produces the value 0, completely disregarding the results of the other similarities. Given that it is more likely to find no overlap between concepts than finding a perfect overlap, it is more reasonable to apply this function on a set of distances instead.

Minkowski Distance

A generalization of the *Euclidean Distance*, the *Minkowski Distance* is defined on a multidimensional Euclidean space. Given a set of n distance metrics $D = \{dist_1, dist_2, \dots, dist_n\}$, a parameter p and two concepts x and y, an aggregation function $a(x, y, S)$ can be defined as a *Minkowski Distance* as follows:

$$a(x, y, D) = \sqrt[p]{\sum_{i=1}^{n} dist_i(x, y)^p} \tag{3.2}$$

Note that choosing $p = 1$ would be equivalent to the *Manhattan distance* while chooseing $p = 2$ would be equivalent to the *Euclidean Distance.*

Weighted Sum

A more linear aggregation can be achieved through the application of a weighted sum. Given a set of n similarity metrics $S = \{sim_1, sim_2, \dots, sim_n\}$, a set of weights $\{\omega_1, \omega_2, \dots, \omega_n\}$ and two concepts x and y, an aggregation function $a(x, y, S)$ can be defined as a weighted sum as follows:

$$a(x, y, S) = \sum_{i=1}^{n} sim_i(x, y) \times \omega_i \tag{3.3}$$

Note that by enforcing a normalization amongst the weights, such that $\sum_{i=1}^{n} =$ 1, one can ensure that the weighted sum is normalized as well. Furthermore, by selecting $\omega_x = \frac{1}{n}, \forall x \in [0 \ldots n]$ one can model a simple arithmetic mean.

Machine Learning

The previously described aggregation techniques have only a limited capability to adapt themselves to the given set of similarity techniques. The weights are typically selected by a domain expert and do not change throughout an entire mapping task. Such a system can have disadvantages for certain scenarios.

To illustrate this, let us return to the example we described at the beginning of this subsection. The example entailed the evaluation of two concept pairs, being $< Bus, Coach >$ and $< Bus, Buss >$, by combining the results of a string similarity and a lexical similarity. Let us assume that the two techniques produce the similarity set $S_1 = \{0, 1\}$ for the pair $< Bus, Coach >$ and the set $S_2 = \{0.9, 0\}$ for the pair $< Bus, Buss >$. Additionally, let us consider a concept pair which should not be matched, being $< Car, Caravan >$ with a resulting similarity set $S_3 = \{0.5, 0.5\}$. Here, it becomes easy to see how no selection of the weights ω would result in an adequate value for all concept pairs. Aggregating both similarities equally would result in S_1, S_2 and S_3 being aggregated into a similar value near 0.5. One would rather desire for the resulting values for S_1 and S_2 to be higher than the value for S_3. Emphasizing the string similarity with appropriate weights for ω would result in S_2 being aggregated into a higher value, but would also lower the result for S_1. The opposite would occur if the lexical similarity were to be emphasized instead.

In order to create a more complex aggregation system capable of taking the intricacies of particular similarity metrics into account, on can apply *machine learning* techniques (Duda, Hart, and Stork, 1999; Russell and Norvig, 2003). In contrast to the previously described aggregation techniques, machine learning approaches require a dataset for each specific application. Such a dataset is typically referred to as the training-set. A machine learning technique attempts to build a model based on the provided training-set such that this model can be used for the purpose of prediction or classification. For the evaluation of a trained machine learning approach one typically uses a separate dataset, referred to as the test-set.

While the term *machine learning* encompasses a broad range of similar problems, for ontology mapping the most relevant techniques are those which tackle *supervised learning*. Here, the training set consists of a large quantity of instances where each instance contains a value for each modelled dimension, i.e. no missing values, and the desired outcome for that instance. This outcome may be a classification label or a numerical value. The applied machine learning technique must construct a model based on the provided instances such that it can approximate the desired result for new instances as closely as possible. For ontology mapping one can model the aggregation task both as a classification and as a regression task (Ichise, 2008). In a classification task the given machine learning approach would assign a label, e.g. *'true'* or *'false'*, to every given concept pair, while in a regression task the technique would assign each concept pair a value in $[0, 1]$. Thus, in order to apply a machine learning technique as an aggregation function, one needs to construct a training-set using the similarity metrics of the mapping system and create a label for each instance. Here one can use ontologies which have already been mapped, for example the datasets of the OAEI campaign. An example of a system using such techniques would be YAM++ (Ngo, Bellahsene, and Coletta, 2012), which utilizes machine learning techniques to aggregate the results of its element-level similarities. Another example system is *NOM/QOM* (Ehrig and Sure, 2004), which uses neural networks for the combination of its matching rules.

We will briefly describe one technique to serve as examples of a machine learning approach. Specifically, we will introduce *artificial neural networks.* For a more thorough introduction to the field of machine learning, we suggest the reader consult the work of Russell and Norvig (2003).

An *artificial neural network* (ANN) is a learning model inspired by the biological structure of the brain (Russell and Norvig, 2003). A brain consists of a large quantity of processing cells, called neurons. Each neuron has a series of incoming and outgoing connections to other neurons. A neuron receives electrical pulses from other neurons through its incoming connections. Based on the presence of absence of pulses a neuron decides whether itself should fire a pulse. This pulse is then forwarded to other neurons through its outgoing connections.

In an ANN the functionality of a neuron is modelled mathematically. A neuron is defined of a series of n input parameters x. Each input is weighted with an individual parameter ω. All weighted inputs are aggregated in a transfer function T, which is typically modelled as $T = \sum_{i-1}^{n} \omega_i \times x_i + b$. The result of the transfer function is used as input for the activation function $A(T, \theta)$, which decides whether the neuron should 'fire' a pulse or not. The activation function may be modelled as a simple 'step' function, with the threshold θ deciding if $A(T, \theta)$ should produce the output 1 or 0. Alternatively, it can be modelled as a continuous function, e.g. a sigmoid function.

While a single neuron can be used for simple learning problems, ANNs typically use a series of interconnected neurons in order to tackle more challenging tasks. The neurons are typically arranged in layers, where a neuron in one layer only receives the output of the previous layer. The output of the system is in essentially defined as a series of nested neuron functions. Figure 3.6 displays an example of a ANN.

Unlike when training a decision tree, the structure of an artificial neural network is already defined before the learning step. Instead, the learning step involves the tuning of all the weights ω, the bias b and the threshold θ for each neuron in the network. A well known algorithm commonly applied for this task is the *backpropagation algorithm* (Rumelhart, Hinton, and Williams, 1986). This algorithm first evaluates the training-set using the current parameters of the network and calculates an error score. By backpropagating the desired outputs through the network a delta score is computed for each neuron. The parameters for each neuron are then updated based on its corresponding delta value. This backpropagation routine is repeated until all instances in the training-set are correctly classified or until a stopping criteria is met.

Once the chosen machine learning technique has a fully trained model using the given input-value pairs, one can use it to predict the appropriate value of new inputs. In the case of an ANN, the similarity values are entered in the input layer and

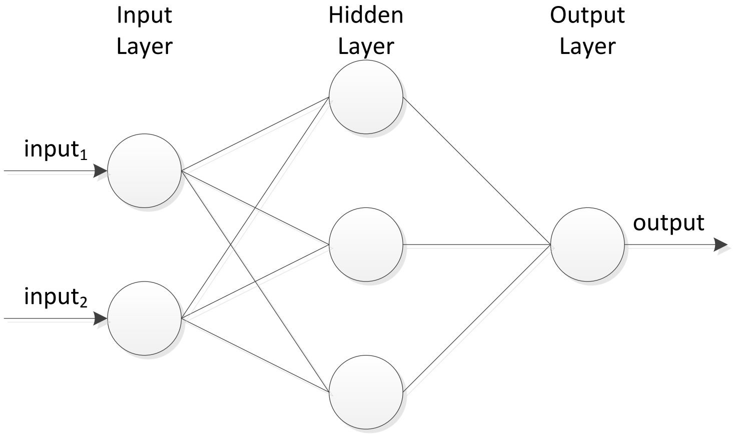


Figure 3.6: Illustration of a neural network.

forwarded through the activation functions $A(T, \theta)$ of each node, using the learned weights ω and the learned bias b for each respective node. The activation function of the output node produces the aggregated similarity value.

3.1.4 Correspondence Extraction

The third core task required for each ontology mapping system is the extraction of correspondences from the aggregated similarities. For each concept c the extraction method must analyse the similarities in the aggregated similarity matrix M and decide to which other concept c should be mapped in a correspondence. Alternatively, the system could decide not to map c at all if the aggregated similarities for c are all sufficiently low below a threshold. Formally, we define the task of correspondence extraction as follows:

Definition 10 (**Correspondence Extraction**). *Given a set of m entities C_1, a set of n entities C_2 and the $m \times n$ similarity matrix M, the task of correspondence extraction is defined as a function $\tau(C_1, C_2, M) \rightarrow A$ which generates an alignment $A \subseteq C_1 \times C_2$.*

The execution of the correspondence extraction function represents the first moment in which an ontology mapping system generates an alignment between the two input ontologies. Based on the system's structure, the alignment can be refined in a post-processing step, used as input alignment for another matcher or simply returned as output of the entire mapping system.

In this subsection we will introduce some common techniques that are used as correspondence extraction methods. These range from simple threshold-based methods

(Do and Rahm, 2002; Ehrig and Sure, 2004) to bipartite matching methods (Euzenat and Shvaiko, 2007).

Hard threshold

In the most straight-forward extraction method, a concept pair is only added to A as a correspondence if their corresponding similarity value in M is larger or equal to a specified threshold θ. This technique can be seen in the *Anchor-PROMPT* (Noy and Musen, 2001) and *QOM* system (Ehrig and Staab, 2004).

Delta threshold

Here the threshold is represented as a delta value d, specifying an absolute or relative tolerance value which is subtracted from the highest retrieved similarity value. The threshold θ is thus specified as $\theta = max(M) - d$. The delta threshold is also referred to as *proportional threshold* if d is specified as a relative value with respect to $max(M)$ (Euzenat and Shvaiko, 2007).

Percentage

In this extraction technique all pairwise combinations of concepts are sorted in descending order. Consequently, only the top $n\%$ of concept pairs are added as correspondences to A.

Relative threshold

Introduced by Melnik, Garcia-Molina, and Rahm (2002) for the evaluation of the *Similarity Flooding* algorithm, this technique splits the similarity matrix M into two matrices M_1 and M_2. The core idea is that the absolute similarity values are transformed into relative similarities with respect to the alternatives of either the concepts of C_1 or the concepts of C_2. Thus, M_1 is generated by normalizing each row r of M using the maximum value of r, and M_2 is generated by normalizing each column k of M using the maximum value of k. The relative threshold technique only selects entity pairs such that their corresponding relative similarities in both M_1 and M_2 are above the specified relative threshold θ.

Let us illustrate this technique using a small example. Given the two entity sets $C_1 = \{a, b\}$ and $C_2 = \{c, d\}$, let us assume that the similarity aggregation technique produced the following similarity matrix M:

$$M = \begin{array}{c} \\ a \\ b \end{array} \begin{array}{c} \begin{array}{cc} c & d \end{array} \\ \begin{pmatrix} 1 & 0.8 \\ 0.54 & 0.27 \end{pmatrix} \end{array}$$

The similarity matrix M is now converted into two matrices M_1 and M_2 specifying relative similarities:

$$M_1 = \begin{array}{c} \\ a \\ b \end{array} \begin{array}{c} \begin{array}{cc} c & d \end{array} \\ \begin{pmatrix} 1 & 0.8 \\ 1 & 0.5 \end{pmatrix} \end{array} \qquad M_2 = \begin{array}{c} \\ a \\ b \end{array} \begin{array}{c} \begin{array}{cc} c & d \end{array} \\ \begin{pmatrix} 1 & 1 \\ 0.54 & 0.33 \end{pmatrix} \end{array}$$

Selecting the threshold $\theta = 0.5$ for example would lead to the entity pairs $< a, c >$, $< a, d >$ and $< b, c >$ being added as correspondences to the output alignment A.

Maximum weight graph matching

Correspondence extraction is another area where techniques from the field of graph theory are of use. Specifically, one can formulate the task of correspondence extraction as a *bipartite graph matching* problem. Here, we are given a graph $G = \{V, E\}$ consisting of the set of vertices V and edges E. G is defined as a bipartite graph if there exist two sets U and W such that $U \cup W = V$, $U \cap W = \emptyset$ and all edges in E only connect vertices from U to W. Additionally, all edges in E are associated with a weight x, such that a specific edge e is defined as a triplet $< v, v', x >$. The task of bipartite graph matching is defined as the identification of a set E' such that $E' \subseteq E$, E' maximizes a certain criteria and no two edges in E' share a common vertex.

One can easily see how one can formulate the task of correspondence extraction as a bipartite graph. First, we create U and W such that each vertex in U represents an entity in C_1 and each vertex in W represents an entity of C_2. Next, the edges in E are specified according to the weights in the similarity matrix M. While for correspondence extraction the vertices in U are fully connected to the vertices in W, bipartite graph matching also considers graphs in which U is sparsely connected to W.

Given a bipartite graph, the core task is now to find a set of edges $E' \subseteq E$ where no two edges in E' share a common vertex. If x_i is defined as the weight of the edge e_i, in *maximum weight matching* one needs to identify an E' such that for every possible $E'' \subseteq E$, where no edge in E'' shares a common vertex, the following inequality holds:

$$\sum_{e_i \in E'} x_i \geq \sum_{e_j \in E''} x_j$$

A well known technique to compute the maximally weighted match is the Hungarian method (Kuhn, 1955). This type of extraction technique is utilized in the *AgreementMaker* system (Cruz, Antonelli, and Stroe, 2009) and *YAM++* system (Ngo *et al.*, 2011).

Stable marriage graph matching

When using this technique the extraction task is also formulated as a bipartite graph matching problem. However, the selection of E' is determined according to a different criterion. Instead of simply maximizing over the weights x, the edges are selected such that a stability criterion is met. A selection E' is considered stable if the following condition holds:

$$\forall < v_1, v_1', x_1 >, < v_2, v_2', x_2 > \in E' : \nexists < v_1, v_2', x > \in E; x \geq x_1 \wedge x \geq x_2$$

More informally expressed, there must exist no pair of vertices v and v' of which the corresponding edge weight in E is higher than the weight of the edges containing v or v' in the selection E'.

The common analogy used here is the coupling between a group of men and a group of women. Each man and woman prepare a sorted list of partners to which they would rather be married to. The goal is to create a list of couples which are to be married. One assumption is that a woman and a man will drop their current partners and select each other if their preferences for each other are higher than their preferences to their current assigned partners. A selection of marriages is considered stable if no unmatched man and woman are willing drop their current assigned partners because they prefer each other more.

A well known algorithm to compute a stable marriage in a bipartite graph is the Gale-Shapley algorithm (Gale and Shapley, 1962). Stable-marriage graph matching is applied in the *MapSSS* system (Cheatham, 2013).

While both *maximum weight matching* and *stable-marriage matching* select a subset of E according to a global criterion, they can produce different results. Let us illustrate their differences using the following example matrix M:

$$M = \begin{array}{c c} & \begin{array}{cc} c & d \end{array} \\ \begin{array}{c} a \\ b \end{array} & \begin{pmatrix} 1 & 0.6 \\ 0.6 & 0.0 \end{pmatrix} \end{array}$$

Since both approaches tackle a bipartite graph matching problem, they can only select two edges out of E since no vertex may be matched multiple times. In this case, two subsets of E are potential outputs for each approach: $E_1 = \{< a, c, 1 >, < b, d, 0 >\}$ and $E_2 = \{< a, d, 0.6 >, < b, c, 0.6 >\}$. *Maximum weight matching* would select E_2, since the sum of the edge weights of E_2, being 1.2, is higher than the sum of edge weights of E_1, being 1. *Stable-marriage matching* however would select E_1, since both a and c prefer each other more than any alternative. *Stable-marriage matching* will only select an E' in which a and c are matched, even if the other vertices are matched to partners with a low weight.

Generally, we can see that *Stable-marriage matching* is more likely to make a selection where a proportion of the edges have a very high weight, with the remaining edges having lower weights due to being matched with lesser preferred vertices. *Maximum weight matching* is more likely to make a more balanced selection, possibly omitting edges with a very high weight.

Naive descending algorithm

The *Naive descending algorithm* presents a greedy approach for the extraction of a 1:1 mapping from a matrix of similarities (Meilicke and Stuckenschmidt, 2007). Instead of maximizing over a global criterion as in the previous approaches, this

Algorithm 3.1 Naive descending algorithm pseudo-code

```
Naive-Descending(M)

$A \leftarrow convertToAlignment(M)$
$sortDescending$(A)
$A' \leftarrow \emptyset$
while $A \neq \emptyset$ do
    $c \leftarrow$ removeFirstElement(A)
    $A' \leftarrow A' \cup c$
    for all $c' \in getAlternatives(A, c)$ do
        removeElement(A,c')
    end for
end while
return $A'$
```

algorithm iterates over the elements of M and extracts correspondences based on a series of local decisions. This extraction technique can be found in the *PRIOR* system (Mao and Peng, 2006).

The algorithms receives as input the similarity matrix M. In the preparatory phase, M is converted into an alignment A such that every entry in M is represented as a correspondence. This operation is denoted using the *convertToAlignment* statement.A is then sorted in descending order according to the confidences of the correspondences. Next, the algorithm iterates over A. In each iteration the correspondence c with the highest confidence value is removed from A and added to the output alignment A'. A is then inspected for any correspondences which contain one of the entities of c, denoted by the *getAlternatives* statement, and removes them from A. This ensures that each concept is mapped at most once. The iteration is continued until no more elements are left in A, at which point A' is returned as the output.

Naive ascending algorithm

The *naive ascending algorithm* functions in a similar way than the naive descending algorithm. However, instead of accepting correspondences and removing alternatives, it dismisses correspondences if there exist alternatives with higher confidence values. This means that a correspondence c will be dismissed if an alternative c' exists with higher confidence value, even if c' is not actually part of the output alignment A'. This makes the naive ascending algorithm more restrictive than the naive descending algorithm, such that *Naive-Ascending*$(M) \subseteq$ *Naive-Descending*(M) for all matrices M (Meilicke and Stuckenschmidt, 2007)[1]. This extraction technique can be found in the *CroMatcher* system (Gulić and Vrdoljak, 2013).

[1] The original authors specify this relation as a sub-set relation ($\subset$). However, their outputs are equal when given a square M containing only non-zero values along the diagonal.

Algorithm 3.2 Naive ascending algorithm pseudo-code

```
Naive-Ascending(M)

$A \leftarrow convertToAlignment(M)$
$sortAscending$(A)
$A' \leftarrow \emptyset$
for all $c \in A$ do
    if $getHigherAlternatives(A, c) = \emptyset$ then
        $A' \leftarrow A' \cup c$
    end if
end for
return $A'$
```

3.2 Mapping system survey

In the previous sections we have introduced the components which form an ontology mapping system and various techniques which can make up the components of such a system . Given the large pool of techniques and the possibility of combining techniques or entire systems, it is evident that there is a vast amount of ways in which one can set up a mapping system. In this section we will provide an overview of a selection of ontology mapping systems. For the sake of brevity, the intention is not to provide an exhaustive list of all systems, but rather to provide an overview of state-of-the-art systems with an emphasis on systems that are related to the topic of this thesis. To get a more complete overview of existing ontology mapping systems, we suggest the reader consults the works by Euzenat and Shvaiko (2007), Rahm and Bernstein (2001), Wache *et al.* (2001), Noy (2004) or the technical reports of the participating systems of the OAEI competition (Aguirre *et al.*, 2012; Grau *et al.*, 2013). We will provide an overview of the different properties of each system in Table 3.1 and a brief introduction to the general mapping strategy of each system.

YAM++

As evidenced by the results of the 2012 Ontology Alignment Evaluation Initiative (Aguirre *et al.*, 2012), one of the current state-of-the-art ontology mapping system is YAM++, developed by (Ngo *et al.*, 2012). This system combines machine learning and information retrieval techniques on the element level and similarity propagation techniques on the structure level to derive a mapping, to which consistency checking is applied to further increase the quality.

AML

The framework AgreementMakerLight (AML) (Cruz *et al.*, 2009; Faria *et al.*, 2013) matches ontologies using a layered approach. In the initial layer, similarity matrices are computed using syntactic and lexical similarities based on WordNet, among others, which are then used to create a set of mappings. Their applied lexical similarity is noteworthy because in the most recent version of AML it included a probabilistic WSD approach based on the assumption that word can be polysemous. (Cruz *et al.*, 2013) Further iterations in subsequent layers refine the existing mappings using structural properties in order to create new mappings. After a sufficient amount of iterations, multiple computed mappings are selected and combined in order to form the final mapping.

ASMOV

ASMOV (Jean-Mary, Shironoshita, and Kabuka, 2009) is capable of using general lexical ontologies, such as WordNet, as well as domain specific ontologies, for instance UMLS, in its matching procedure. After creating a mapping using a set of similarity measures, a semantic verification process is performed in order to remove correspondences that lead to inferences which cannot be verified or are unlikely to be satisfied given the information present in the ontologies.

Anchor-PROMPT

The PROMPT tool has been a well known tool in the field of ontology mapping (Noy and Musen, 2003). Based on the Protégé environment, the tool features various ontology editing features, such as ontology creation, merging and versioning. It also features a user-based mapping process in which it iterates between querying the user for mapping suggestions and using the user input to refine the mapping and formulate new queries. A notable extension to PROMPT is referred to as Anchor-PROMPT (Noy and Musen, 2001). This is a mapping algorithm which utilizes partial alignments. The algorithm traverses all possible paths between two anchors in both ontologies and records which concept pairs are encountered at the same time. The intuition here is that concept pairs which are more frequently encountered during the traversal steps are more likely to denote the same entity.

S-Match

The S-Match suite (Giunchiglia, Shvaiko, and Yatskevich, 2004) is a notable example for its semantic mapping approach, thus being one of the first mapping systems utilizing a semantic mapping technique. Particularly, it employs the JSAT SAT solver in order to derive concept correspondences based on a set of initial correspondences. These initial correspondences are generated using a set of element-level similarities, e.g. language and corpus-based techniques.

PRIOR

One of the first formal definitions of a profile similarity was developed by Mao *et al.* (2007), the researchers behind the PRIOR system. Their definition of a concept profile is limited to the information encoded in its definition. Thus information of related concepts is excluded in that definition. However, related information is added in a method which they refer to as *profile propagation.* To improve efficiency for large mapping tasks, the system can omit the *profile propagation* step and derive concept similarities using the basic profiles and information retrieval-based indexation methods (Mao and Peng, 2006).

Falcon-AO

Falcon-AO is another system with an early development of profile similarities (Jian *et al.*, 2005; Hu and Qu, 2008). The authors use the term 'virtual document' to describe the idea of a profile, emphasizing the origin of the approach from the field of information retrieval (Qu *et al.*, 2006). Their profile creation model is noteworthy because it facilitates parametric weighting of the profile terms according to their original source.

LogMap

LogMap is an example of a mapping system which generates a set of anchors during the mapping process (Jiménez-Ruiz and Cuenca Grau, 2011; Jiménez-Ruiz *et al.*, 2012b). This is done by efficiently comparing labels in a pre-computed index of the given ontologies. The computed anchors are exploited in a mapping discovery and repairs phase. The system alternates between discovering new correspondences and discarding unsatisfiable correspondences that have been identified in the previous step. This ensures that new correspondences are logically sound such that the resulting alignment is consistent. New correspondences are discovered by using the I-SUB string metric (Stoilos, Stamou, and Kollias, 2005).

AUTOMS

The AUTOMS system (Kotis, Valarakos, and Vouros, 2006a) is noteworthy due to its implementation of the HCONE-Merge approach (Kotis, Vouros, and Stergiou, 2006b). This approach assumes the existence of an intermediate hidden ontology and attempts to create it using the information of both given ontology, effectively merging the input ontologies. To determine whether two concepts should be merge into one definition, the system establishes whether a linguistic match and/or a lexical match exists. For the lexical matcher, concepts are disambiguated using latent semantic indexing (LSI). The system also employs structural information if there is no sufficient syntactic meta-information to proceed with the HCONE-Merge approach.

Anchor-FLOOD

Anchor-FLOOD is a partial alignment-based mapping tool inspired by the Anchor-PROMPT system (Seddiqui and Aono, 2009). However, instead of exploring paths between anchors, it utilizes the anchors as a starting point for the discovery of new mappings. In an iterative procedure, the system repeatedly selects a single anchor to explore. The surrounding concepts of the anchor are analysed using terminological and structural similarities. Concept pairs which satisfy a threshold criteria are added to the alignments and the iterative process is repeated until no more correspondences can be discovered.

NOM/QOM

NOM (Naive Ontology Mapping) (Ehrig and Sure, 2004) is a heuristic mapping system, which follows a series of codified rules specifying what meta-information of each entity type can be used to derive a similarity value. The system supports 17 decision rules which have been specified by domain experts. An example of such a rule would be *R9*, stating that two properties are similar if their sub-relations are similar. *QOM*(Quick-Ontology-Mapping) (Ehrig and Staab, 2004) is a variation of the NOM system with an added emphasis on computational efficiency. This allows the system to tackle large-scale problems. To achieve the intended efficiency QOM restricts several features of the NOM system. For instance, concept trees are compared in a top-down approach instead of computing the full pairwise similarities of the trees.

RiMOM

RiMOM is an example of an ontology mapping system which can automatically adapt itself to better the fit the task at hand (Li *et al.*, 2009). Prior to mapping, it calculates two factors, the *label-similarity-factor* and the *structure-similarity-factor*, over both ontologies. These factors describe the similarity of the entire given ontologies based on their entity-name overlap and structural overlap. The values of these measures determine the selection of the applied similarity measures, how these are tuned and influence the similarity aggregation step. The resulting similarities are improved by propagating them through a pairwise-connectivity-graph (PCG) using the Similarity Flooding algorithm (Melnik *et al.*, 2002).

AOT

AOT is a recently developed system that participated in the 2014 edition of the OAEI competition (Khiat and Benaissa, 2014). It uses a combination of multiple string similarities and a lexical similarity in order to derive its concept correspondences.

WeSeE

WeSeE is an example how the internet can be used as background knowledge for the mapping procedure. The system constructs a document for each concept by querying the search engine *Bing* using the terms from the concept labels and comments as queries (Paulheim, 2012). The search results are then processed, merged and weighted using TF-IDF weighting, such that the concept similarity is then defined as the similarity between their corresponding documents.

MapSSS

MapSSS is another example of a system utilizing search engines. However, in contrast to WeSeE it does not assemble a document using the search results. Instead, queries are formulated using the concept labels and specific keywords, such as ‘translation’ or ‘synonym’. The similarity score is them determined by the amount and quality of the retrieved results (Cheatham, 2011; Cheatham, 2013).

COMA++

COMA++ (Aumueller *et al.*, 2005) is a refined mapping system of its predecessor COMA (Do and Rahm, 2002). It stands out from other systems due to its application of alignment re-use and fragment-based matching techniques. The system can be operated by a user using a GUI and completes the mapping task in user-activated iterations. Before each iteration, the user can provide feedback in the form of confirming correspondences or refuting correspondences of previous iterations. The user can also decide which alignment-paths to exploit for the alignment re-use technique or let the system decide automatically based on the criterion of expected effort.

CroMatcher

CroMatcher is a recently developed system consisting of mainly syntactical methods (Gulić and Vrdoljak, 2013). The general matching strategy is arranged in a sequential order in two components. The first component evaluates concept pairs using string, profile, instance and internal-structural similarities. Instead of extracting an alignment and forwarding this to the second component, the system forwards the entire similarity matrix. The second component uses the matrix to initialize several structural similarities, comparing entities with respect to the super-entities, sub-entities, property domains or ranges. The alignment is extracted using the Naive-Ascending algorithm.

WikiMatch

WikiMatch (Hertling and Paulheim, 2012a; Hertling and Paulheim, 2012b) a noteworthy variant of the WeSeE system. However, instead of exploiting search-engines using heuristically generated queries, it exploits the contents of *Wikipedia* as an external resource. For every term in a concept label, fragment or comment the set of corresponding Wikipedia article is retrieved. The system defines the similarity between two concepts as the maximum Jaccard similarity that can be attained by comparing the Wikipedia article sets of two terms.

System	YAM++	AML	ASMOV	Anchor-PROMPT	S-Match
Input	OWL	XSD,RDFS,OWL	OWL	OWL	XML
User-interaction	-	-	-	editing,mergins, versioning, matching	-
Architecture	iterative	sequential, iterative	iterative	hybrid	basic
Syntactic (Ele.)	string, profile	string	string, internal structure	string, internal structure	string, language-based
Syntactic (Stru.)	external profile, shared instances	profile	taxonomy structure, instance similarity	taxonomy	label-profile co-occurrence
Lex. Resources	-	external ontology	WordNet,UMLS	WordNet	WordNet
Lex. Resources (WSD)	-	probabilistic	-	-	-
Alignment Re-use	-	-	-	-	-
Partial Alignments	-	-	-	anchor traversal	-
Rep. of Structures	-	-	-	-	-
Semantic (Stru.)	-	-	-	-	SAT solver
Remarks	machine-learning aggregation	-	semantic verification as stopping criterion	-	-

Table 3.1: Overview of ontology mapping systems.

System	PRIOR	Falcon-AO	LogMap	AUTOMS	Anchor-FLOOD
Input	OWL	RDFS, OWL	OWL	OWL	OWL
User-interaction	-	GUI operation	matching	-	-
Architecture	basic	basic	sequential,iterative	basic	iterative
Syntactic (Ele.)	string	string	string, language-based	string	string,internal structure
Syntactic (Stru.)	profile	profile, graph-based	-	taxonomy	taxonomy
Lex. Resources	-	-	synonym extraction	-	synonym overlap
Lex. Resources (WSD)	-	-	-	HCONE-Merge, LSI	-
Alignment Re-use	-	-	-	-	-
Partial Alignments	-	-	generated, anchor-based mapping discovery	-	anchor-based mapping discovery
Rep. of Structures	-	-	-	-	-
Semantic (Stru.)	-	-	-	-	-
Remarks	complexity-based similarity selection	PBM partitioning for large tasks	structural and lexical indexation	-	-

Table 3.1: (Continued) Overview of ontology mapping systems.

System	NOM/QOM	RiMOM	AOT	WeSeE	MapSSS
Input	RDFS, OWL	OWL	OWL	OWL	OWL
User-interaction	-	-	-	-	-
Architecture	iterative	basic	basic	basic	iterative
Syntactic (Ele.)	string, internal structure	string	string	-	string
Syntactic (Stru.)	taxonomy	profile	-	-	graph-based
Lex. Resources	application-specific vocabulary	-	WordNet	Bing	GoogleResearch
Lex. Resources (WSD)	-	-	-	-	-
Alignment Re-use	-	-	-	-	-
Partial Alignments	-	-	-	-	-
Rep. of Structures	-	-	-	-	-
Semantic (Stru.)	-	-	-	-	-
Remarks	-	similarity propagation, strategy selection & tuning	-	concept profiles using search engines	-

Table 3.1: (Continued) Overview of ontology mapping systems.

System	COMA++	CroMatcher	WikiMatch
Input	SQL,XSD, OWL	OWL	OWL
User-interaction	GUI, user feedback, re-use selection	-	-
Architecture	iterative	sequential	basic
Syntactic (Ele.)	string, internal structure	string, internal structure	-
Syntactic (Stru.)	NamePath, children and leaves overlap	profile, instances parent & child overlap	-
Lex. Resources	synonyms	-	Wikipedia
Lex. Resources (WSD)	-	-	-
Alignment Re-use	alignment paths	-	-
Partial Alignments	-	-	-
Rep. of Structures	fragment matching	-	-
Semantic (Stru.)	-	-	-
Remarks	-	-	-

Table 3.1: (Continued) Overview of ontology mapping systems.

Chapter 4

Concept-Sense Disambiguation for Lexical Similarities

This chapter is an updated version of the following publications:

1. Schadd, Frederik C. and Roos, Nico (2011a). Improving ontology matchers utilizing linguistic ontologies: an information retrieval approach. *Proceedings of the 23rd Belgian-Dutch Conference on Artificial Intelligence (BNAIC 2011)*, pp. 191−198.
2. Schadd, Frederik C. and Roos, Nico (2011b). Maasmatch results for OAEI 2011. *Proceedings of The Sixth International Workshop on Ontology Matching (OM-2011) collocated with the 10th International Semantic Web Conference (ISWC-2011)*, pp. 171−178.
3. Schadd, Frederik C. and Roos, Nico (2012a). Coupling of WordNet Entries for Ontology Mapping using Virtual Documents. *Proceedings of The Seventh International Workshop on Ontology Matching (OM-2012) collocated with the 11th International Semantic Web Conference (ISWC-2012)*, pp. 25−36.
4. Schadd, Frederik C. and Roos, Nico (2014b). Word-Sense Disambiguation for Ontology Mapping: Concept Disambiguation using Virtual Documents and Information Retrieval Techniques. *Journal on Data Semantics*, Zimàyi, Esteban, Ram, Sudha and Stuckenschmidt Heiner ed., pp. 1−20, Springer.

As seen in subsection 3.2, there exist various mapping systems which exploit external resources to derive concept correspondences. Of these, lexical and ontological resource-based techniques are the most popular. For each entity, these techniques allocate a set of entries within the lexical resource or ontological resource. If an entity of a resource is encoded as a set of synonyms instead of a named concept, then it may also be referred to as a synonym-set (synset). Lexical similarities then compare two concepts by evaluating their correspondence between sets of senses within the resource. The most basic techniques simply compute the overlap of the two sets (Ehrig and Sure, 2004; Seddiqui and Aono, 2009). While these techniques can effectively establish the equivalence between two entities based on their synonym-relation within the resource, they provide no differentiation between concept pairs which are closely related and pairs which are not related at all. For instance, given two ontologies O_1 and O_2, where O_1 contains the concept *Car* as the only vehicle-related concept and O_2 only contains *Vehicle* as a transport related concept. In this situation, one would be inclined to map *Car* with *Vehicle* as there are no better alternatives for either concept. However, since *Car* and *Vehicle* are not synonymous, their resulting lexical similarity would be 0 when applying a sense-overlap method.

To be able to establish the lexical similarity between non-synonymic concepts, it is necessary to apply a more sophisticated evaluation methods. Examples of such methods are the comparison of the sets of leaf-nodes of each sense in the taxonomy or their relative distance within the resource. However, an inherent property of a natural language is that words can have multiple senses. This means that for an ontology concept it may be the case that there are multiple possible entries in the exploited external resource. The accuracy of a lexical similarity thus depends on whether the senses which denote the correct meaning are actually exploited.

This chapter answers the first research question by proposing an information-retrieval-based disambiguation method. We extend existing glossary-overlap based methods by using a profile-based method of creating concept descriptions, as introduced in subsection 3.1.2. The proposed profile method gathers terms into a virtual document, facilitating the inclusion of terms of related concepts and the weighting of terms according to their origin. We evaluate the effects of various sense-selection policies and lexical similarity metrics by analysing the alignment quality when matching the OAEI benchmark and conference datasets. Further, we evaluate the effect of different weighting policies and quantify the introduced overhead of the disambiguation policy.

This chapter is structured as follows. Section 4.1 introduces important background information regarding lexical similarities, word-sense disambiguation techniques and virtual documents. Section 4.2 discusses previous work relating to the content of this chapter. Section 4.3 introduces the framework in which concept terms are disambiguated and lexical similarities are established. Section 4.4 presents the performed experiments and discusses its results and 4.5 gives the conclusion of this chapter and discusses future research.

4.1 Background

In this section we will provide background knowledge which will be necessary for the remainder of this chapter. In subsection 4.1.1 we will introduce lexical similarity measures in more detail. Since the problem posed by *research question 1* addresses the task of disambiguating ontology concepts, we will introduce the reader to the field of word-sense disambiguation in subsection 4.1.2. Finally, we will introduce virtual documents in subsection 4.1.3 such that the reader is adequately prepared for a virtual document-based disambiguation method, which will be introduced in Section 4.3.

4.1.1 Lexical Similarity Measures

Lexical similarity measures (LSM) are commonly applied metrics in ontology mapping systems. These exploit externally available knowledge bases which can be modelled in ontological or non-ontological form, for instance by utilizing databases. Such a knowledge base contains a list of concepts describing the particular domain that is being modelled. Each concept description contains various kinds of information, such as synonyms and written explanations of that concept. If such a description does contain at least a list of synonyms, it is also often referred to as *synset* (synonym-set). Another important feature of a knowledge base is that each concept is also linked to other concepts using various semantic relations, thus creating a large relational structure. A LSM exploits these large structures by linking ontology concepts to the nodes in the external knowledge base, such that the proximity of concepts associated with source and target concepts provides an indication to their similarity. One can here distinguish between *semantic relatedness* and *semantic similarity* (Strube and Ponzetto, 2006), where semantic relatedness denotes the measured proximity by exploring all given relations, and semantic similarity expresses the proximity using only *is-a* relations. Whether a LSM determines *relatedness* or *similarity* depends on the utilized metric which expresses the proximity (Budanitsky and Hirst, 2001; Giunchiglia and Yatskevich, 2004) since the definitions of these metrics typically also define which relations are exploited. For this research, as further detailed in sub-section 4.3.2, the applied metric utilizes only *is-a* relations, rendering the base LSM which our approach intends to improve, as a measure of semantic similarity.

There exist several lexical knowledge bases which can be used as a resource for a LSM. These originate from different research efforts and were all developed with different capabilities, which can roughly be grouped as follows:

Global/Cross-Domain Knowledge Resources of this category intend to model a multitude of domains, such that the similarity between concepts can be identified even if these are generally categorized in different domains. A prime example of such a resource is WordNet, which models the domain of the entire English language into approximately 120.000 interrelated synonym sets (Miller, 1995). This resource is regularly used in contemporary ontology mapping systems. WordNet is also available as an extended version in the form of YAGO (Suchanek, Kasneci, and Weikum, 2008), which merges WordNet with the concept descriptions available from Wikipedia.

Domain Knowledge These resources intend to model common knowledge of a single specified domain. Typically, these domains are not very broadly defined, however they are usually modelled in great detail. A collaborative effort for the creation of a knowledge resource is Freebase, which contains both general knowledge and named entities and is available in both database and ontological form (Bollacker *et al.*, 2008). UMLS (Bodenreider, 2004) is a good example how detailed a domain ontology can become. It is a biomedical resource which models 900.000 concepts using 2 million labels by integrating and interlinking several existing vocabularies.

Abstract Upper Ontology Resources belonging to this group have the singular focus of creating an abstract ontology using an upper-level list of concept descriptions. Such an ontology can then serve as a base for domain specific resources. An example of such a resource is the SUMO ontology, containing approximately 2.000 abstract concept descriptions (Niles and Pease, 2001). These concepts can then be used to model more specific domains. MILO for instance is an extension of SUMO which includes many mid-level concepts (Niles and Terry, 2004). Cyc is another example of a multi-layered ontology based on an abstract upper level-ontology (Matuszek *et al.*, 2006), of which a subset is freely available under the name OpenCyc (Sicilia *et al.*, 2004).

Multi-lingual When mapping ontologies, it can occur that some concept descriptions are formulated in a different language. In these situations mono-lingual resources are insufficiently applicable, necessitating the usage of multi-lingual resources, e.g. UWN (De Melo and Weikum, 2009) or BabelNet (Navigli and Ponzetto, 2010).

LSMs are a powerful metric and are commonly used in contemporary state-of-the-art ontology mapping systems (Shvaiko and Euzenat, 2005; Kalfoglou and Schorlemmer, 2003; Saruladha, Aghila, and Sathiya, 2011), with WordNet being the most widely used resource as basis. However, a common occurrence in concepts formulated using natural language is word-sense ambiguity. This entails that a word can have multiple and possibly vastly different meanings, such that one must eliminate all meanings which do not adequately represent the intended meaning of the word. This task, while at a glance quite intuitive for a human, can be deceptively difficult for a computer program. Given the word *house* for instance, the intended meaning might be obvious to a human reader, however this word has 14 different meanings listed in WordNet, such that an accurate identification of the correct sense is necessary in order to obtain accurate results. The histogram in Figure 4.1 indicates the extent of such situations occurring within WordNet (Miller, 1995). Here, all unique words that occur in WordNet have been gathered and binned according to how many different meanings each word describes.

One can see from Figure 4.1 that while there is a large number of words with only one meaning, there is a significant proportion of words which do have more than one meaning and hence can ambiguous. The general working hypothesis is that a word in a given context has only a single correct sense. The rejection of this hypothesis, the acknowledgement of polysemous words, is an emerging field of research for which new approaches are emerging (Cruz *et al.*, 2013). Ultimately a LSM has to calculate the similarity between two sets of senses, where the assumption whether these sets can contain multiple correct senses may influence the choice of specific employed techniques, including disambiguation methods. LSMs can incorporate polysemous concepts by for instance calculating an aggregate similarity between these sets of senses (Euzenat and Shvaiko, 2007; Cruz *et al.*, 2013; Po and Sorrentino, 2011). However, if a domain expert determines that the concepts in the ontology are not polysemous, one can adapt the aggregation step by for instance only utilizing the maximum pairwise similarity (Euzenat and Shvaiko, 2007) between sets of senses or by selecting the predominant sense as determined by a given corpus (McCarthy *et al.*, 2004). The inclusion of a word-sense disambiguation technique in a LSM, which this chapter proposes, is likely to improve their accuracy.

4.1.2 Word-Sense Disambiguation

Word-Sense Disambiguation (WSD) can be described as the automatic identification of the correct sense(s) of a given word using the information in the proximity of that word as context. While in many works only one sense is associated with each word, we define WSD as a process which selects a set of possible candidate senses. The resulting set may contain multiple senses if desired by the expert designing the system, for instance to accommodate polysemous words. In the classical problem of disambiguating words occurring in natural language, the available context information is a body of text co-occurring with the target word (Navigli, 2009). Depending on the input document or the applied approach, this body of context information can be limited to the sentence in which the target word appears or extended over the entire input document. The available context information originating from an ontology is different compared to a natural language document. In an ontology, natural language is a rare occurrence and usually limited to brief concept descriptions in the form of annotations. Hence, context information must be extracted from the entire concept description, its associated properties and other related concepts.

Originally, WSD has been perceived as a fundamental task in order to perform

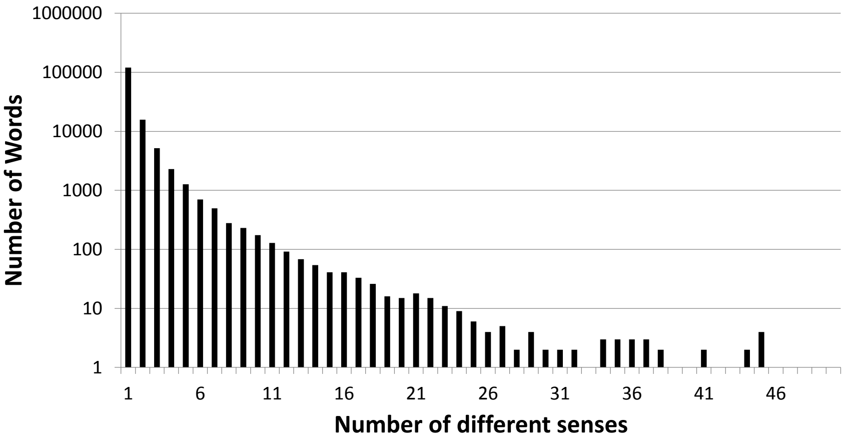


Figure 4.1: Histogram showing the number of words in WordNet (y-axis) that have a specific number of senses (x-axis).

machine translation (Weaver, 1955; Locke and Booth, 1955). Here, the establishment of accurate word-senses is a requirement for the selection of correct word translations from a multi-lingual dictionary. While research into WSD halted for a decade after its acknowledged hardness (Bar-Hillel, 1960), it has been re-instigated after Wilkes (1975) tackled this problem using formal semantics in order to achieve a computer understanding of natural language. For a more comprehensive overview of the history of WSD we suggest the reader consults the work of Ide and Véronis (1998).

Many different approaches to WSD have been developed over the past decades. Due to the prevalence of applied machine-learning techniques, three general categories of approaches have emerged:

Supervised Disambiguation One can formulate WSD as a classification problem. Here, a training set is creating by tagging sentences with the correct senses of its contained words. Once the training set has reached a sufficient size, one can use this as basis for a supervised classification method. Examples of such methods are decision lists, decision trees, Naive Bayes classifier, Neural-Networks, instance-based methods such as the kNN approach and ensemble methods which combine different classifiers (Montoyo *et al.*, 2005; Navigli, 2009).

Unsupervised Disambiguation These methods have the advantage that they do not rely on the presence of a manually annotated training set, a situation which is also referred to as the knowledge acquisition bottleneck (Gale, Church, and Yarowsky, 1992). However, unsupervised methods share the same intuition behind supervised methods, which is that words of the same sense co-occur alongside the same set of words (Pedersen, 2006). These rely on clustering methods where each cluster denotes a different word sense.

Knowledge-based Disambiguation Instead of applying classification techniques, knowledge-based methods exploit available knowledge resources, such as dictionaries, databases or ontologies, in order to determine the sense of a word (Mihalcea, 2006). These techniques are related to LSMs in that they often exploit the same knowledge resources. This group of techniques will be further discussed in subsection 4.2.1.

For a more comprehensive survey of disambiguation techniques we suggest the reader consults the excellent survey by Navigli (2009).

While originally conceived for the purpose of machine translation, WSD techniques have been applied in a variety of tasks (Ide and Véronis, 1998). In the field of information retrieval, one can apply WSD in order to eliminate search results in which at least some of the query keywords occur, but in a different sense than the given query (Schütze and Pedersen, 1995). This would lead to a reduction of false positives and hence increase the performance of the retrieval system.

WSD can also aid in the field of content and thematic analysis (Litkowski, 1997). Here, the aim is to classify a given text into thematic categories, such as traditional (e.g. judicial, religious), practical (e.g. business), emotional (e.g. leisure, fiction) and analytical (e.g. science) texts. Given a corpus of training data, one can create a profile for each defined category consisting of the distributions of types of words over a text.

In the field of grammatical analysis WSD is required in order to correctly identify the grammatical type of ambiguous words (Marshall, 1983). WSD can also aid a speech synthesis system such that ambiguous words are phoneticised more accurately (Sproat, Hirschberg, and Yarowsky, 1992). Yarowsky (1994) applied WSD techniques for text processing purposes with the aim to automatically identify spelling errors.

4.1.3 Virtual Documents

The general definition of a virtual document (VD) (Watters, 1999) is any document for which no persistent state exists, such that some or all instances of the given document are generated at run-time. These stem from an emerging need for documents to be more interactive and individualized, which is most prominently seen on the internet. A simple form of a virtual document would be creating a template for a document. While some of its content is static, the remainder needs to be filled in from a static information source, for instance a database. This is commonly applied in government agencies which sent automated letters, in which relevant data such as the recipients information is added to the letter template. Composite documents are a different type of virtual document. Here, the content of several documents are combined and presented to the user as a single document. A virtual document can also contain meta-data that has been collected from various documents. Commonly, this can be seen in review-aggregation websites, such as Metacritic, IMDb and RottenTomatoes, which automatically query many independent review sources and aggregate their results into an overall consensus.

In the domain of lexical similarity metrics the basic data structure used for the creation of a virtual document is a linked-data model. It consists of different types of binary relations that relate concepts, i.e. a graph. RDF (Lassila, Swick, and W3C, 1998) is an example of a linked-data model, which can be used to denote an ontology according to the OWL specification (McGuinness and Van Harmelen, 2004). The inherent data model of a thesaurus such as WordNet has similar capacities, however it stores its data using a database. A key feature of a linked-data model is that it not only allows the extraction of literal data for a given concept, but also enables the exploration of concepts that are related to that particular concept, such that the information of these neighbouring concepts can then be included in the virtual document. From the linked-data resource, information is gathered and stored in a document with the intention that the content of that document can be interpreted as a semantic representation of the meaning of a specific ontology concept, which in turn can be exploited for the purpose of ontology mapping. A specific model for the creation of such a virtual document will be presented in subsection 4.3.3.

4.2 Related Work

4.2.1 Methods of Word-Sense Disambiguation

There exists a notable spectrum of word-sense disambiguation techniques, which have been used for varying purposes, however certain techniques stand out due to their applicability to this domain. The method of context clustering (Schütze, 1992) can be used to exploit large amounts of labelled training data. Here, co-occurrences with a target word are modelled as a vector in a word space and grouped into clusters according to their labelled word-sense. Given a new occurrence of the given word, one can identify its sense by modelling a new context vector form its neighbourhood and classifying it using the created word-sense clusters. This can be done for instance by determining the centroid vector of each cluster and computing the vector distance for each centroid vector.

A more linguistic approach can be achieved through the application of selectional preferences (Hindle and Rooth, 1993). By determining the grammatical types of words within a sentence, one can limit the amount of possible sense by imposing limitation according to the grammatical or semantic context of a particular word. Such a technique can be especially relevant for the mapping of ontology properties, since property names or labels can contain combination of grammatical types, e.g. nouns, verbs or adjectives, where its proper classification can improve their semantic annotations.

A very effective group of disambiguation methods are those based on glossary overlap, which are a knowledge-based methods. These rely on the presence of a detailed corpus of word senses that include their descriptions in natural language. Determining the overlap between the set of words occurring in context of a target word and the different sense-descriptions of that word within the given corpus can be used to determine its proper sense. This type of method has been pioneered by Lesk (1986), which can be improved by incorporating the descriptions of words that are related to the different possible senses (Banerjee and Pedersen, 2003).

Cross-lingual word-sense disambiguation is another knowledge-based approach which exploits multilingual corpora (Resnik and Yarowsky, 1999). A target word is translated into several distinct languages such that the intended sense is likely the one whose meaning has been preserved for the majority of the used languages.

Structural methods (Pedersen, Banerjee, and Patwardhan, 2005) exploit the concept structure of a given corpus. This is achieved by applying a similarity metric between word senses, such that the disambiguated sense of a word from a text is the particular sense which maximizes the aggregate similarities between all possible senses of the words occurring in the text and itself.

Budanitsky and Hirst (2001) evaluated five different sense-similarity measures which serve as the basis for structural disambiguation methods, however these are also applicable to lexical similarities between ontology concepts. For a more in-depth survey of word-sense disambiguation methods, especially the types which do not strongly relate to the techniques applied in this research, we suggest the reader consult the comprehensive survey by Navigli (2009).

4.2.2 Word-Sense Disambiguation in Ontology Mapping

Given the large set of possible techniques originating from many different research areas that can be applied to the process of ontology mapping, only limited research has been performed into applying word-sense disambiguation techniques. Some of this research involves the creation of annotation frameworks, which can facilitate a standardized format of lexical concept annotations and can even provide a more fine-grained annotation. An example of such a framework is the work of Buitelaar *et al.* (2009), who proposed a linguistic labelling system for the annotation of ontology concepts. While the primary intent of this system was the facilitation of ontology learning and natural language generation from ontologies, the linguistic meta information of this system can also be used to disambiguate word-senses, for instance by extracting selectional preferences generated from these annotations.

McCrae, Spohr, and Cimiano (2011) proposed a common model for linking different lexical resources to ontology concepts. This model not only includes constructs modelling terms and their senses, but also the morphosyntactic properties of terms which would allow for a more fine grained annotation of ontology concepts.

Some ontology mapping systems apply WSD to aid their lexical similarity measures. The AUTOMS system (Kotis *et al.*, 2006a), which is designed for the task of ontology merging, employs a technique called HCONE-Merge (Kotis *et al.*, 2006b). Part of this technique involves the process of *latent semantic indexing* (LSI), which is used to associate senses with the given ontology concepts. The approach assumes that concepts are monosemous. Ontology concepts are associated with the sense which resulted in the highest score when querying a latent semantic space using a binary query. This space is created by performing singular value decomposition on the sense descriptions.

Po and Sorrentino (2011) introduced a probabilistic WSD method which has been included in the *AgreementMaker* system (Cruz *et al.*, 2013). Here, each ontology concept is annotated with a set of possible senses, where each sense is annotated with a probability value. This probability value is determined by combining the results of several WSD techniques, i.e. structural disambiguation, domain disambiguation and first-sense heuristic, using the Dempster-Shafer Theory. This method is related to our work due to its application of the basic *Lesk* method as one of the different WSD techniques. The approach of our paper also relies on the principle behind the *Lesk* method, such that substituting our approach for the basic *Lesk* method could improve the WSD accuracy of the *AgreementMaker* system.

A search-engine-based disambiguation method has been proposed by Maree and Belkhatir (2014). They calculate a distance between a concept and a synset by calculating the normalized retrieval distance (NRD). This measure utilizes the separate search-engine retrieval rates of two entities and their co-occurrence rate to compute a normalized co-occurrence distance. A synset s is only associated with a concept c if their NRD value $NRD(s, v)$ satisfies a manually specified threshold.

4.3 Concept Sense Disambiguation Framework

Our proposed approach aims at improving matchers applying lexical similarity metrics. For this resarch, the applied lexical similarity measuref will use WordNet as knowledge-resource. The synsets of WordNet will be used to annotate the meanings of ontology concepts and express their semantic relatedness.

The goal of our approach is to automatically identify the correct senses for each concept of an ontology. This will be realized by applying information retrieval techniques on virtual documents that have been created using either ontology concepts or word sense entries from the knowledge resource. To achieve this, we need to define a lexical similarity in which we can integrate a disambiguation procedure. First, let us define the sets E_1 and E_2, originating from the ontologies O_1 and O_2 respectively, as the sets of entities that need to be matched. These can be classes, properties and/or instances, though these three categories of entities are typically matched separately. Also, these may represent the complete sets of classes/properties within the ontologies or just subsets in case a partitioning method has been applied. Next, let us denote e as an entity and $S(e)$ as the set of senses representing e. Furthermore let us denote lsm as a lexical similarity that can be invoked after the disambiguation procedure and ϕ as a disambiguation policy. We define our lexical similarity as specified in Algorithm 4.1 [1].

The initial step of the approach, denoted as the method *findSynsetCandidates*, entails the allocation of synsets that might denote the meaning of a concept. The name of the concept, meaning the fragment of its URI, and alternate labels, when provided, are used for this purpose. While ideally one would prefer synsets which contain an exact match of the concept name or label, precautions must be made for the eventually that no exact match can be found. For this research, several pre-processing methods have been applied such as the removal of special characters, stop-word removal and tokenization. It is possible to enhance these precautions further by for instance the application of advanced natural language techniques, however the investigation of such techniques in this context is beyond the scope of this research. When faced with ontologies that do not contain concept names using natural language, for instance by using numeric identifiers instead, and containing no labels, it is unlikely that any pre-processing technique will be able to reliably identify possible synsets, in which case a lexical similarity is ill-suited for that particular matching problem.

In the second step, the virtual document model as described in section 4.3.3 is applied to each ontology concept and to each synset that has been gathered in the previous step. This procedure is denoted as *createVD* in the algorithm. The resulting virtual document are represented using the well known vector-space model (Salton, Wong, and Yang, 1975). In order to compute the similarities between the synset documents and the concept documents, the established cosine-similarity is applied (Pang-Ning, Steinbach, and Kumar, 2005). Using the specified filtering policy ϕ synsets are discarded whose cosine-similarities of their documents to the concept document did not satisfy the criteria specified by ϕ. This process is denoted as *discardDocuments* in the pseudo-code. The different criteria which ϕ can represent

[1] $E_1[i]$ denotes the i-th element of E_1

Algorithm 4.1 Lexical similarity with disambiguation pseudo-code

```
Lexical-similarity-WSD($E_1$,$E_2$,$\phi$,$lsm$)

for all $e \in E_1 \cup E_2$ do
    $S(e) \leftarrow findSynsetCandidates(e)$
    $doc(e) \leftarrow createVD(e)$
    $D \leftarrow \emptyset$
    for all $s \in S(e)$ do
        $doc(s) \leftarrow createVD(s)$
        $D \leftarrow D \cup \{doc(s)\}$
    end for
    $D \leftarrow discardDocuments(D, doc(e), \phi)$
    for all $s \in S(e)$ do
        if $doc(s) \notin D$ then
            $remove(s, S(e))$
        end if
    end for
    $assignSynsets(e, S(e))$
end for
$M \leftarrow initMatrix(|E_1|, |E_2|)$
for $i = 1$ to $|E_1|$ do
    for $j = 1$ to $|E_2|$ do
        $e_1 \leftarrow E_1[i]$
        $e_2 \leftarrow E_2[j]$
        $M[i, j] \leftarrow lsm(e_1, e_2)$
    end for
end for
return $M$
```

will be introduced in the following subsection.

In the following sub-sections we will elaborate important components of Algorithm 4.1 in more detail. In subsection 4.3.1 we will detail the disambiguation policies which utilizes the resulting document similarities, denoted as *discardDocuments* in Algorithm 4.1. Subsection 4.3.2 details the task of the lexical similarity function lsm and the tested variations. In subsection 4.3.3 we will introduce the utilized document model and finally in subsection 4.3.4 we will briefly introduce the TF-IDF weighting method that can be applied to the documents.

4.3.1 Concept Disambiguation

Once the similarities between the entity document and the different synset documents are known, a selection method ϕ is applied in order to disambiguate the meaning of the given concept. Here, senses are only coupled to the concept if they resulted in a sufficiently high document similarity, while the remaining senses are discarded. To determine which similarity score can be considered sufficiently high, a selection policy needs to be applied. It is possible to tackle this problem from various angles, ranging from very lenient methods, discarding only the very worst synsets, to strict methods, associating only the highest scoring synset with the given concept (Banek, Vrdoljak, and Tjoa, 2008). Several selection methods have been investigated, such that both strict and lenient methods are tested:

G-MEAN The most lenient method aggregates the document similarities using the geometric mean and uses this as a threshold to discard senses with a lower similarity value.

A-MEAN Similar to the previous method, however the arithmetic mean is used as a threshold instead.

M-STD This more strict method dynamically determines a threshold by subtracting the standard deviation of the document similarities from the highest obtained similarity. It has the interesting property that it is more strict when there is a subset of documents that is significantly more similar than the remaining documents, indicating a strong sense correspondence, and more lenient when it not as easy to identify the correct correspondences.

MAX The most strict method consists of dismissing all senses from C except for the one single sense that resulted in the highest document similarity.

Once all concepts of both input ontologies are disambiguated, one can compute the lexical similarity between concepts using the processed synset collections.

4.3.2 Lexical Similarity Metric

After selecting the most appropriate synsets using the document similarities, the similarity between two entities can now be computed using their assigned synsets. This presents the problem of determining the similarity between two sets of synsets.

To approach this task, we will evaluate three different methods of determining the lexical similarity between two collections of synsets.

A reasonable assumption is that each collection of synsets only contains one synset that represents the true meaning of its corresponding entity. Thus, if one were to compare two sets of synsets, assuming that the originating entities are semantically related, then one can assume that the resulting similarity between the two synsets that both represent the true meaning of their corresponding entities, should be a high value. Inspecting all pairwise similarities between all combinations of synsets between both sets should yield at least one high similarity value. When comparing two sets originating from semantically unrelated entities, one can assume that there should be no pairwise similarity of high value present. Thus, in this scenario a reasonable way of computing the similarity of two sets of synsets is to compute the maximum similarity over all pairwise combination between the two sets. This intuition is similar to the principle of Maximum Relatedness Disambiguation (Pedersen *et al.*, 2005) in the sense that two concepts can be considered similar if a certain amount of their concept information can be considered similar by some measure.

Formally, given two concepts x and y, their corresponding collections of synsets $S(x)$ and $S(y)$, a measure of semantic similarity $sim(m,n) \in [0,1]$ where m and n are two arbitrary synsets, we define the first lexical similarity lsm_1 between x and y as:

$$lsm_1(x,y) = \max_{m \in S(x); n \in S(y)} sim(m,n) \tag{4.1}$$

The work by Gao, Zhang, and Chen (2015) serves as an example how lsm_1 can be used to compute the similarity between sets of senses. A potential weakness of lsm_1 is the eventuality where the a concept has several appropriate senses. When comparing these senses to other collections, one might prefer a method which values the quantity of high similarities as well. For example, assume that the sense collections $S(x)$, $S(y)$ and $S(z)$ each contain two senses and that we wish to determine whether $S(y)$ or $S(z)$ are a more appropriate match for $S(x)$. Further, assume that each pairwise similarity between $S(x)$ and $S(y)$ results in the value ψ, where as only one pair of senses from $S(x) \times S(z)$ results in the similarity ψ with the remaining pairs being unrelated and resulting in a similarity of 0. Computing $lsm_1(x,y)$ and $lsm_1(x,z)$ would both result in the value ψ. In this example however, one would be more inclined to match x with y since the comparison with y resulted in more high similarity values. A way to adapt for this situation is to determine the best target sense for each sense in both collections and to aggregate these values, which we will denote as lsm_2. Given two concepts x and y, their corresponding collections of synsets $S(x)$ and $S(y)$, a measure of semantic similarity $sim(m,n) \in [0,1]$ where m and n are two arbitrary synsets, we define lsm_2 as follows:

$$lsm_2(x,y) = \frac{\sum_{m \in S(x)} (\max_{n \in S(y)} sim(m,n)) + \sum_{n \in S(y)} (\max_{m \in S(x)} sim(n,m))}{|S(x)| + |S(y)|} \tag{4.2}$$

A more general approach to determine the similarity between two collections of senses is to aggregate all pairwise similarities between the two collections. This has the potential benefit that similarity values which have no effect on the result of lsm_1 or lsm_2 are affecting the outcome of the lexical similarity measure. We will denote this measure as lsm_3. Formally, given two concepts x and y, their corresponding collections of synsets $S(x)$ and $S(y)$, a measure of semantic similarity $sim(m, n) \in [0, 1]$, we define lsm_3 as follows:

$$lsm_3(x, y) = \frac{\sum_{m \in S(x)} \sum_{n \in S(y)} sim(m, n)}{|S(x)| \times |S(y)|} \tag{4.3}$$

There exist various ways to compute the semantic similarity sim within WordNet (Budanitsky and Hirst, 2001) that can be applied, however finding the optimal measure is beyond the scope of this paper since this is not a component of the disambiguation process. Here, a similarity measure with similar properties as the Leacock-Chodorow similarity (Budanitsky and Hirst, 2001) has been applied. The similarity $sim(s_1, s_2)$ of two synsets s_1 and s_2 is computed using the distance function $dist(s_1, s_2)$, which determines the distance of two synsets inside the taxonomy, and the over depth D of the taxonomy:

$$sim(s_1, s_2) = \begin{cases} \frac{D - dist(s_1, s_2)}{D} & \text{if } dist(s_1, s_2) \leq D \\ 0 & \text{otherwise} \end{cases} \tag{4.4}$$

This measure is similar to the Leacock-Chodorow similarity in that it relates the taxonomic distance of two synsets to the depth of the taxonomy. In order to ensure that the resulting similarity values fall within the interval of $[0, 1]$ and thus can be integrated into larger mapping systems, the log-scaling has been omitted in favor of a linear scale.

4.3.3 Applied Document Model

We will provide a generalized description of the creation of a virtual document based on established research (Qu *et al.*, 2006). The generalization has the purpose of providing a description that is not only applicable to an OWL/RDF ontology like the description given in the work by Qu *et al.* (2006), but also to non-ontological knowledge sources. While a variety of external resources can be utilized, for this research we will use the most widely utilized resource, which is WordNet (Miller, 1995). To provide the functions that are used to create a virtual document, the following terminology is used:

Synset: Basic element within a knowledge source, used to denote a specific sense using a list of synonyms. Synsets are related to other synsets by different semantic relations, such as hyponymy and holonymy.

Concept: A named entity in the linked-data model. A concept denotes a named class or property given an ontology, and a synset when referring to WordNet.

Link: A basic component of a linked-data model for relating elements. A link is directed, originating from a source and pointing towards a target, such that the type of the link indicates what relation holds between the two elements. An example of a link is a triplet in an RDF graph.

sou(s), type(s), tar(s): The source element, type and target element of a link s, respectively. Within the RDF model, these three elements of a link are also known as the subject, predicate and object of a triplet.

Collection of words: A list of unique words where each word has a corresponding weight in the form of a rational number.

+: Operator denoting the merging of two collections of words.

× Operator denoting multiplication.

A concept definition within a linked-data model contains different types of literal data, such as a name, different labels, annotations and comments. The RDF model expresses some of these values using the *rdfs:label*, *rdfs:comment* relations. Concept descriptions in WordNet have similar capacities, but the labels of a concepts are referred to as its synonyms and the comments of a concept are linked via the glossary relation.

> **Definition** Let ω be a concept of a linked-data model, the description of ω is a collection of words defined by (4.5):
>
> $$\begin{aligned} Des(\omega) \quad = \quad & \alpha_1 \times \text{collection of words in the name of } \omega \\ & +\alpha_2 \times \text{collection of words in the labels of } \omega \\ & +\alpha_3 \times \text{collection of words in the comments of } \omega \\ & +\alpha_4 \times \text{collection of words in the annotations of } \omega \end{aligned} \tag{4.5}$$

Where each α_1, α_2, α_3 and α_4 is a rational number in $[0, 1]$, such that words can be weighed according to their origin.

Next to accumulating information that is directly related to a specific concept, one can also include the descriptions of neighbouring concepts that are associated with that concept via a link. Such a link can be a standard relation that is defined in the linked-data model, for instance the specialization relation and also a ontology-defined property if the used syntax allows the property to occur as a predicate. While theoretically the presented model would also allow instances to be included if these are present in the ontology, it is very unlikely that a given knowledge resource contains similar specific instance information for which an overlap can be determined. Hence, given instances are filtered from the ontologies before the creation of the documents. The OWL language supports the inclusion of blank-node concepts which allow complex logical expressions to be included in concept definitions. However, since not all knowledge resources support the blank-node functionality, meaning anonymous concepts defined using a property restriction, among which WordNet, these are omitted in our generalization. For more information on how to include blank nodes in the description, consult the work by Qu *et al.* (2006).

To explore neighbouring concepts, three neighbour operations are defined. $SON(\omega)$ denotes the set of concepts that occur in any link for which ω is the source of that link. Likewise $TYN(\omega)$ denotes the set of concepts that occur in any link for which ω is the type, or predicate, of that link and $TAN(\omega)$ denotes the set of concepts that occur in any link for which ω is the target. WordNet contains inverse relations, such as hypernym being the inverse of the hyponym relation. When faced with two relations with one being the inverse of the other, only one of the two should be used such that descriptions of neighbours are not included twice in the virtual document. The formal definition of the neighbour operators is given below.

> **Definition** Let ω be a named concept and s be a variable representing an arbitrary link. The set of source neighbours $SON(\omega)$ is defined by (4.6), the set of type neighbours $TYN(\omega)$ of ω is defined by (4.7) and the set of target neighbours $TAN(\omega)$ of ω is defined by (4.8).

$$SON(\omega) = \bigcup_{sou(s)=\omega} \{type(s), tar(s)\} \tag{4.6}$$

$$TYN(\omega) = \bigcup_{type(s)=\omega} \{sou(s), tar(s)\} \tag{4.7}$$

$$TAN(\omega) = \bigcup_{tar(s)=\omega} \{sou(s), type(s)\} \tag{4.8}$$

Given the previous definitions, the definition of a virtual document of a specific concept can be formulated as follows.

> **Definition** Let ω be a concept of a linked-data model. The virtual document of ω, denoted as $VD(\omega)$, is defined by (4.9):

$$\begin{aligned} VD(\omega) = &Des(\omega) + \beta_1 \times \sum_{\omega' \in SON(\omega)} Des(\omega') \\ &+ \beta_2 \times \sum_{\omega' \in TYN(\omega)} Des(\omega') + \beta_3 \times \sum_{\omega' \in TAN(\omega)} Des(\omega') \end{aligned} \tag{4.9}$$

Here, β_1, β_2 and β_3 are rational numbers in $[0,1]$. This makes it possible to allocate a different weight to the descriptions of neighbouring concepts of ω compared to the description of the concept ω itself.

We will provide a brief example for the resulting term weights in a virtual document that is created using this model. For this we will use an example ontology provided in Figure 4.2.

Suppose one would want to construct a virtual document representing the concept *Car*. The term weights of this document are determined through the merger of the description of the concept *Car* and the weighted descriptions of the concepts *Vehicle* and *Ambulance*. The term weight of the word *car* would be α_1, since the term only occurs in the name of the concept *Car*. The term *vehicle* would receive the weight $\alpha_3 + \beta_1 \times \alpha_1 + \beta_3 \times \alpha_3$. This is because the term occurs in three locations in the neighbourhood of the concept *Car*, once in a comment of the given concept, once in

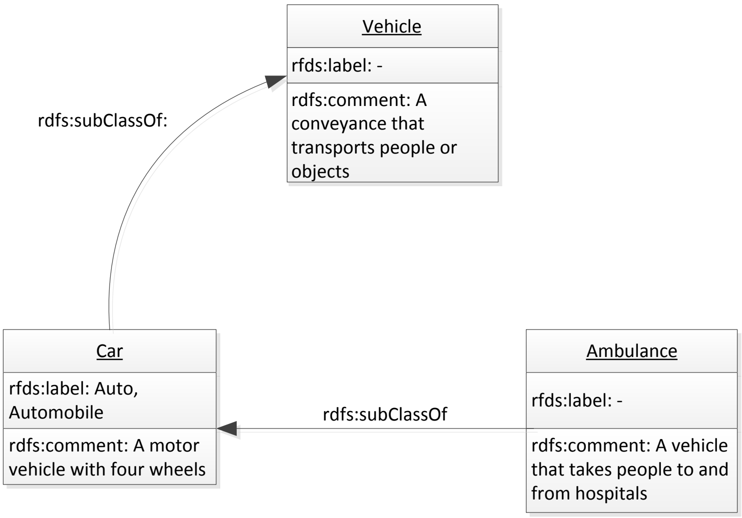

Vehicle
rfds:label: -
rdfs:comment: A conveyance that transports people or objects
rdfs:subClassOf:
Car
rfds:label: Auto, Automobile
rdfs:comment: A motor vehicle with four wheels
rdfs:subClassOf
Ambulance
rfds:label: -
rdfs:comment: A vehicle that takes people to and from hospitals

Figure 4.2: Example ontology for the construction of a virtual document.

the name of a source neighbour and once in a comment of a target neighbour. The sum of these particular occurrences hence forms the final term weight for this word. The full list of term weights of the document representing the example concept *Car* can be viewed in Table 4.1. For the sake of demonstration the list also includes the weights of stop-words.

Term	Weight	Term	Weight
a	$\alpha_3 + \beta_1 \times \alpha_3 + \beta_3 \times \alpha_3$	motor	α_3
ambulance	$\beta_3 \times \alpha_1$	objects	$\beta_1 \times \alpha_3$
auto	α_2	people	$\beta_1 \times \alpha_3 + \beta_3 \times \alpha_3$
automobile	α_2	takes	$\beta_3 \times \alpha_3$
car	α_1	that	$\beta_1 \times \alpha_3 + \beta_3 \times \alpha_3$
conveyance	$\beta_1 \times \alpha_3$	transports	$\beta_1 \times \alpha_3$
four	α_3	to	$\beta_3 \times \alpha_3$
from	$\beta_3 \times \alpha_3$	vehicle	$\alpha_3 + \beta_1 \times \alpha_1 + \beta_3 \times \alpha_3$
hospitals	$\beta_3 \times \alpha_3$	wheels	α_3

Table 4.1: Term weights for the document representing the concept *Car*, according to the example ontology displayed in Figure 4.2.

4.3.4 Term-Frequency Weighting

Instead of weighting terms in a virtual document according to their origin from within their respective ontology, it is possible to treat a virtual document as a standard natural language document once all of its dynamic content has been determined. This allows for the application of well-known weighting techniques originating from the field of information retrieval.

Information retrieval techniques have been applied in a variety of fields. The most prominent application is the retrieval of relevant documents from a repository, as seen in commercial search engines (Croft, Metzler, and Strohman, 2009). Document vectors can be weighed using different methods (Salton and Buckley, 1988), of which the most prominent method is the application of TF-IDF weights (Sparck Jones, 1972). This method relates the term frequency (TF) of a word within a document with the inverse document frequency (IDF), which expresses in how many of the registered documents a term occurs. Given a collection of documents D and an arbitrary term t, the inverse document frequency of term t is computed as follows:

$$idf(t, D) = \log \frac{|D|}{|\{d \in D : t \in d\}|} \tag{4.10}$$

Given the term frequency of the term t within document d_x as $tf(t, d_x)$, the TF-IDF weight of the term t within document d_x is then specified as follows.

$$tf\text{-}idf(t, d_x, D) = tf(t, d_x) \times idf(t, D) \tag{4.11}$$

This weighting scheme assigns higher weights to terms that occur more frequently in a document d_x, however this effect is diminished if this term occurs regularly in other documents as well. The resulting weighed vectors can then be used in a similarity calculation with a query, such that the document that is the most similar to the query can then be seen as the most relevant document.

Given the availability of ontological background knowledge which can aid the document creation process, it is questionable whether the application of a weighting scheme which is designed to be applied on texts formulated in natural language outperforms the weighting functionality supplied by the virtual document model. For this work, we will empirically evaluate the benefits of the TF-IDF weighting scheme when applied to virtual documents using the methods described in section 4.3.3.

4.4 Experiments

In this section, the experiments that have been performed to test the effectiveness of adding a concept disambiguation step, specifically our approach, to a lexical similarity will be presented. These experiments serve to evaluate different aspects of the proposed approach and to demonstrate the feasibility of word-sense disambiguation techniques for an ontology mapping system. The different experiments can be divided into the following categories:

- Subsection 4.4.1 describes the performed experiments to evaluate the different concept disambiguation policies in order to determine whether lenient or strict policies should be preferred.
- The experiments described in subsection 4.4.2 demonstrate the potential performance a system can achieve when utilizing the proposed techniques.
- Subsection 4.4.3 presents the performed experiments which evaluate the considered virtual document weighting techniques.
- The runtime performance overhead and gains introduced by our approach will be analysed in subsection 4.4.4.

The tested mapping system used for the performed experiments contains two similarity metrics: a lexical similarity using a configuration which is specified in the experimental set-up and a syntactic similarity using the Jaro string similarity (Jaro, 1989) applied on concept names and labels. The combined concept similarities are aggregated using the Naive descending extraction algorithm (Meilicke and Stuckenschmidt, 2007). The tested system in sections 4.4.1, 4.4.2 used the parameter schemes obtained from the experiment presented in section 4.4.3, while the system in section 4.4.2 had a manually tuned parameter set. To quantify the quality of the produced alignments we will evaluate them using the measures of thresholded precision, recall and F-measure, as introduced in Section 2.2, with the exception of sub-section 4.4.3 which evaluates the tested approaches using a precision-recall graph and therefore computes the standard measures of precision an recall.

4.4.1 Concept Disambiguation

To investigate to what extent disambiguation techniques can improve a framework using a lexical similarity, we evaluated our approach using different variations of our approach on the conference data set of the 2011 competition (Euzenat *et al.*, 2011c) from the Ontology Alignment Evaluation Initiative (OAEI) (Euzenat *et al.*, 2011b). This data set consists of real-world ontologies describing the conference domain and contains a reference alignment for each possible combination of ontologies from this data set. We performed this evaluation using the three lexical similarity measures lsm_1, lsm_2 and lsm_3, evaluating each measure using the disambiguation policies *G-Mean*, *A-Mean*, *M-STD* and *MAX*. We denote *None* as the omission of the disambiguation step, such that its results denotes the baseline performance of the respective lexical similarity measure. Figure 4.3 displays the different results when using lsm_1.

From Figure 4.3 we can make several key observations. First, we can see that a stricter disambiguation policy clearly benefits the lsm_1 metric, evidenced by the steadily increasing F-measure. The low precision for lenient policies implies that there are numerous false positives which exhibit a higher semantic similarity than the true correspondences. When increasing the strictness of the filtering policy, the precision rises steadily, meaning an increasing amount of false positives are eliminated. We can also observe a slight drop in recall for stricter policies, particularly

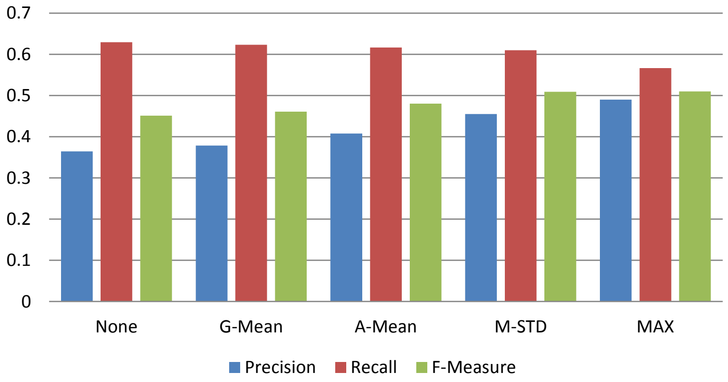


Figure 4.3: Evaluation of disambiguation policies using the lexical similarity lsm_1 on the OAEI 2011 Conference data set.

when comparing *M-STD* with *MAX*, which implies that in a few situations the wrong senses are filtered out.

The same evaluation has been performed using the lsm_2 lexical similarity. The results of this evaluation can be seen in Figure 4.4.

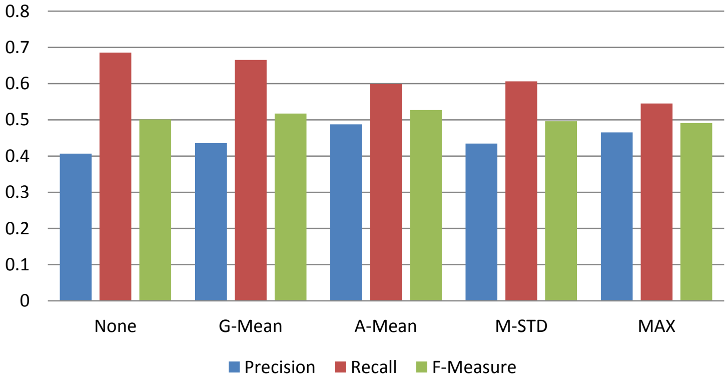


Figure 4.4: Evaluation of disambiguation policies using the lexical similarity lsm_2 on the OAEI 2011 Conference data set.

From Figure 4.4 we can see that the disambiguation policies have a different effect on lsm_2, as opposed to lsm_1. We can observe an improvement in performance when applying *G-Mean* or *A-Mean* as policies, with F-measures of .517 and .526 respectively compared to the baseline F-measure of .501. This improvement stems from an increase in precision, which more than compensates for the loss in recall. However, the F-measure decreases again when applying *M-STD* and *MAX* as policies. This implies that preferring to match concepts whose sense have multiple high pairwise similarities can be beneficial, since for *M-STD* and *MAX* it is unlikely at least that after the disambiguation step there are multiple senses left. Thus, main observation of this evaluation that a disambiguation step is also beneficial for lsm_2, though not for all disambiguation policies.

Lastly, the results of the evaluation when applying lsm_3 can be observed in Figure 4.5.

From Figure 4.5 we can see that the precision and recall values obtained by applying lsm_3 differ significantly when compared to the values obtained by applying lsm_1 or lsm_2. For the baseline and the policies *G-Mean* and *A-Mean* we can observe a very high precision and low recall. The high precision implies that a high average semantic similarity between collections of synsets is likely to represent a true correspondences. The low recall implies though that this does not occur very frequently. Upon applying the most lenient disambiguation policy *G-Mean*, we can see a drastic increase in both recall and F-measure. Applying the stricter policy *A-Mean* the recall and F-measure increases slightly, though at the cost of a reduced precision. The performance of *M-STD* is similar to its performance when applying lsm_1 or lsm_2, implying that it is not a regular occurrence that this policy retains more than one word sense.

Overall, we can conclude that the application of the proposed disambiguation method benefited the tested lexical similarity metrics. For lsm_1 and lsm_3 a strict disambiguation policy has produced the best results, while for lsm_2 the lenient policies have been shown to be most effective.

4.4.2 Framework Comparison

In this subsection we will compare the performance of a mapping system utilizing our approach with the performance of established techniques. To do this, we have entered a configuration of our approach in the OAEI 2011 competition (Euzenat *et al.*, 2011c), of which the results are reported in section 4.4.2. A comparison with the performances of additional and revised state-of-the-art systems will be presented in

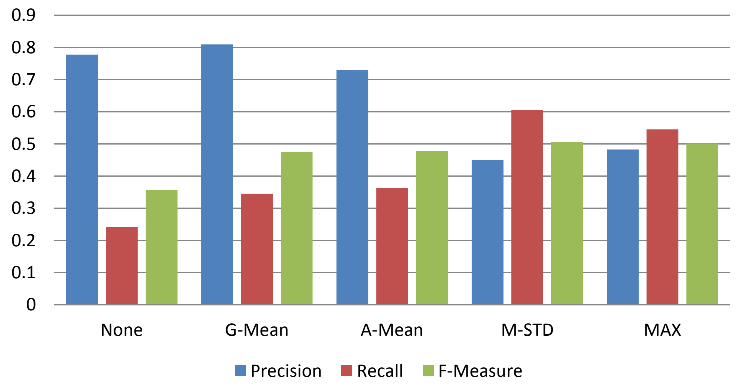


Figure 4.5: Evaluation of disambiguation policies using the lexical similarity lsm_3 on the OAEI 2011 Conference data set.

section 4.4.2.

Preliminary OAEI 2011 evaluation

During the research phase of this approach, we entered the described system in the 2011 OAEI competition under the name *MaasMatch* in order evaluate its performance. The configuration used the lexical similarity metric lsm_1 with disambiguation policy MAX, since at the time the performance of lsm_2 and lsm_3 were not evaluated, yet. The results of the competition on the conference data set can be seen in Figure 4.6.

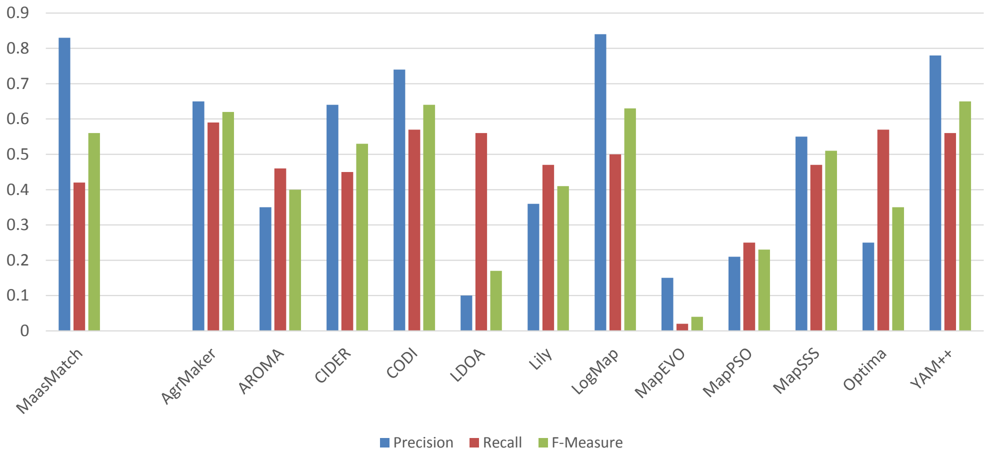


Figure 4.6: Results of MaasMatch in the OAEI 2011 competition on the conference data set, compared against the results of the other participants

From Figure 4.6 one can see that MaasMatch achieved a high precision and moderate recall over the conference data set, resulting in the fifth-highest F-measure among the participants, which is above average. A noteworthy aspect of this result is that this result has been achieved by only applying lexical similarities, which are better suited at resolving naming conflicts as opposed to other conflicts. This in turn also explains the moderate recall value, since it would require a larger, and more importantly a more varied set of similarity values, to deal with the remaining types of heterogeneities as well. Hence, it is encouraging to see these good results when taking into account the moderate complexity of the framework.

A different dataset of the OAEI competition is the *benchmark* data set. This is a synthetic data set, where a reference ontology is matched with many systematic variations of itself. These variations include many aspects, such as introducing errors or randomizing names, omitting certain types of information or altering the structure of the ontology. Since a base ontology is compared to variations of itself, this data set does not contain a large quantity of naming conflicts, which our approach is targeted at. However, it is interesting to see how our framework performs when faced with every kind of heterogeneity. Figure 4.7 displays the results of the OAEI 2011 evaluation (Euzenat *et al.*, 2011c) on the benchmark data set.

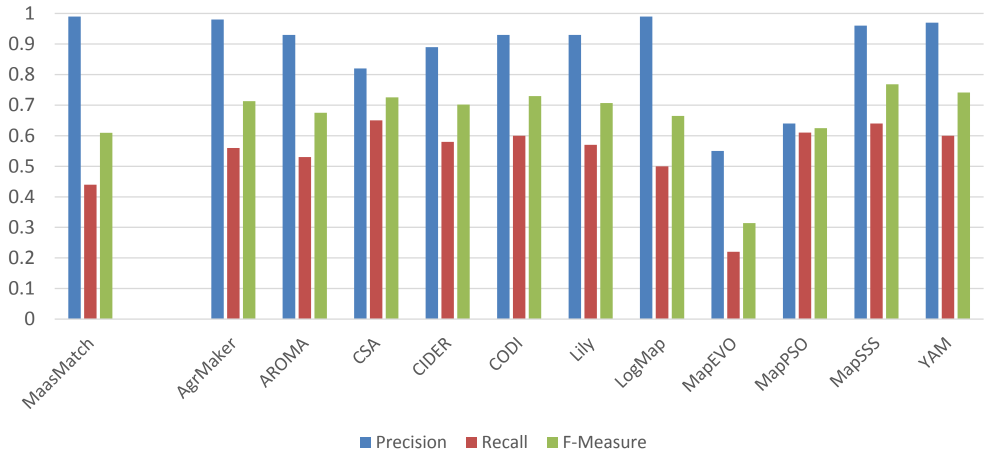


Figure 4.7: Results of MaasMatch in the OAEI 2011 competition on the benchmark data set, compared against the results of the other participants

From Figure 4.7 we can see that the overall performance MaasMatch resulted in a high precision score and relatively low recall score when compared to the competitors. The low recall score can be explained by the fact that the disambiguation method relies on collecting candidate synsets using information stored in the names of the ontology concepts. The data set regularly contains ontologies with altered or scrambled names, such that it becomes extremely difficult to allocate candidate senses which can be used for the disambiguation step. These alterations also have a negative impact on the quality of the constructed virtual documents, especially if names or annotations are scrambled or completely left out, resulting in MaasMatch performing poorly in benchmarks that contain such alterations. Despite these drawbacks, it was possible to achieve results similar to established matchers that address all types of heterogeneities. Given these results, the performance can be improved if measures are added which tackle other types of heterogeneities, especially if such measures increase the recall without impacting the precision.

Comparison with OAEI 2013 frameworks

To give a more complete picture of the performance of our approach compared to other frameworks, we re-evaluated our approach using the 2013 conference data set (Grau *et al.*, 2013) using the same evaluation methodology than the OAEI competition. This allows for the comparison with newer frameworks. Here, the frameworks *edna* and *StringEquiv* are purely string-based systems which serve as a baseline comparison. We limit the comparison to systems which performed above the lowest baseline, *StringEquiv*, for the sake of brevity. We test three variations of our approach, allowing each lexical similarity metric to be compared. As disambiguation policies we applied *MAX* for lsm_1 and *A-Mean* for lsm_2 and lsm_3. While *A-Mean* is sub-optimal for lsm_3 with respect to the F-measure, applying its best-performing measure *MAX* would result in a performance similar to the configuration of lsm_1. The comparison of the OAEI 2013 performances with the three lexical similarity measures can be seen in Table 4.2.

Framework	Precision	Recall	F-Measure
YAM++	.78	.65	.71
AML-bk	.82	.53	.64
LogMap	.76	.54	.63
AML	.82	.51	.63
ODGOMS1_2	.7	.55	.62
StringsAuto	.74	.5	.6
ServOMap_v104	.69	.5	.58
MapSSS	.77	.46	.58
ODGOMS1_1	.72	.47	.57
$\mathbf{lsm_1}$	.8631	.4436	.5685
$\mathbf{lsm_2}$	.7382	.4797	.5643
HerTUDA	.7	.46	.56
WikiMatch	.7	.45	.55
WeSeE-Match	.79	.42	.55
IAMA	.74	.44	.55
HotMatch	.67	.47	.55
CIDER_CL	.72	.44	.55
edna	.73	.44	.55
$\mathbf{lsm_3}$	.6327	.5041	.5466
OntoK	.72	.43	.54
LogMapLite	.68	.45	.54
XMapSiG1_3	.68	.44	.53
XMapGen1_4	.64	.45	.53
SYNTHESIS	.73	.41	.53
StringEquiv	.76	.39	.52

Table 4.2: Evaluation on the conference 2013 data set and comparison with OAEI 2013 frameworks.

One can observe from Table 4.2 that of the three tested lexical similarity measures, lsm_1 and lsm_2 scored above the two baseline matchers. The quality of the alignments produced by the two variants of the tested systems is very similar, especially with respect to the F-measure. Similar to its 2011 performance, the lsm_1 variant displayed a strong emphasis on precision, while the precision and recall of lsm_2 resembles the measures obtained by similarly performing systems, most notably *ODGOMS1_1* and *HerTUDA*. The performance of lsm_3 is more comparable to the *baseline* and the *OntoK* system.

Overall, we can conclude that a system using our approach can perform competitively with state-of-the-art systems, especially when taking into account the modest complexity of the tested system.

4.4.3 Weighting Schemes Experiments

In this section, we will demonstrate the effect of the parameter system of the used document model. We will demonstrate this effect when the model is used to calculate word sense scores, as described in our approach, and the effect when the model is used in its original context as a profile similarity.

Preliminaries: Parameter Optimization

The applied VD model provides the possibility of parametrized weighting, which allows the emphasis of words depending on their origin. Recall from subsection 4.3.3 that the model contains a set of parameters, being α_1, α_2, α_3, α_4, β_1, β_2 and β_3, which weight terms according to their place in the ontology. Next to evaluating the weighting approaches in the proposed WSD method, we will also test a profile similarity that uses the presented virtual document model for gathering the context information of a concept, similar to the work by Qu *et al.* (2006). Here, given two concepts c and d, originating form different ontologies, and their respective virtual documents $VD(c)$ and $VD(d)$, a profile similarity can be created by computing the document similarity between $VD(c)$ and $VD(d)$. For each of the tested approaches the *conference* and *benchmark* datasets were used as separate training sets, resulting in four different parameter sets. We will use the terms Lex-B and Lex-C to refer to the parameter sets which have been generated by optimizing the LSM on the *benchmark* and *conference* dataset respectively. For the parameter sets which have been generated using the profile similarity we will use the terms Prof-B and Prof-C.

Tree-Learning-Search (TLS) (Van den Broeck and Driessens, 2011) was applied in order to optimize the different combinations of similarity metrics and training sets. TLS combines aspects of Monte-Carlo Tree Search and incremental regression tree induction in order to selectively discretize the parameter space. This discretized parameter space in then sampled using the Monte-Carlo method in order to approximate the optimal solution. The results of the performed optimization can be seen in Table 4.3.

Parameter Set	α_1	α_2	α_3	α_4	β_1	β_2	β_3
Lex-C	.51	.68	.58	.42	.32	.07	.06
Lex-B	.52	.99	.08	.65	.01	.09	.16
Prof-C	.71	.02	.01	.58	.09	.04	.01
Prof-B	.85	.13	.54	.32	.90	.32	.99

Table 4.3: Optimized parameter sets for the VD model when applied to a LSM (Lex) and profile similarity (Prof) using the conference (C) and benchmark (B) data sets as training sets.

From Table 4.3 some notable differences emerge. The parameter α_1 tends to have a higher value for profile similarities compared to the LSM parameters sets. This can be explained by the fact that the synset candidate collection step of the proposed disambiguation method selects candidate synsets using the processed ontology concept names as basis. Hence, all sysnet candidates will contain terms that are similar to the ontology concept name, diminishing their information value for the purpose of WSD. Conversely, values for α_2 tend to be higher for LSM parameter sets, indicating that matching alternative concept names are a strong indication of a concept's intended meaning.

Preliminaries: Test Setup

We will evaluate six different weighting schemes for virtual documents in order to investigate what impact these have on the mapping quality. The six weighting schemes were evaluated on the *conference* dataset and can be described as follows:

TF As a reference point, we will evaluate the performance of standard term-frequency weights as a baseline, which is done by setting all VD parameters to 1.

Lex-C/Prof-C This scheme represents the VD model using optimized parameters that were obtained from the same dataset. This scheme will be referred to by the name of its corresponding parameters set, which is Lex-C for the WSD evaluation and Prof-C for the profile similarity evaluation.

Lex-B/Prof-B Similar to the previous scheme, however the parameter set were obtained through the optimization on a different training set.

TF-IDF This scheme entails the combination of term-frequency and inverse document-frequency weights, as commonly seen in the field of information retrieval. Similar to TF weighting, all weights of the VD model will be set to 1.

Lex-C/Prof-C * TF-IDF It is possible to combine the VD model with a TF-IDF weighting scheme. This scheme represents such a combination using the parameter sets that have been obtained from the same data set. In the WSD experiment this scheme will be referred to as Lex-C * TF-IDF, while in the profile similarity experiment it will be referred to as Prof-C * TF-IDF.

Lex-B/Prof-B * TF-IDF Similar to the previous scheme, however the parameter sets that were obtained from the benchmark dataset are used instead.

The evaluation of the TF-IDF method and its combination with the VD model weighting is especially critical since previous work using this model has included TF-IDF weighting in its approach without evaluating the possible implications of this technique (Qu *et al.*, 2006). For each weighting method the computed alignments are ranked according to their similarity. For each ranking the interpolated precision values will be computed such that these can be compared.

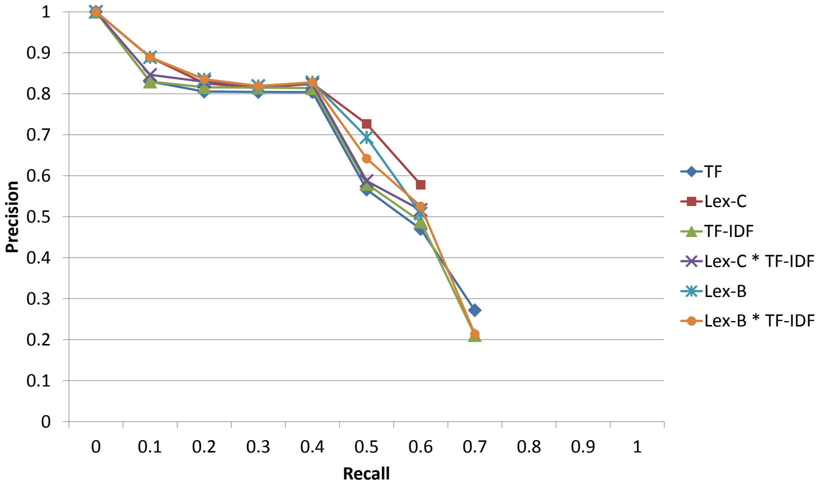


Figure 4.8: Precision versus Recall graph of the created alignments from the conference data set using the lexical similarities with the virtual document

Lexical Similarity with Applied WSD

The different weighting schemes have been separately applied to this approach and subsequently used to calculate mappings on the conference data set. The precision vs recall graph of the produced alignments can be seen in Figure 4.8.

From Figure 4.8 we can observe some key points. For lower recall values, Lex-C, Lex-B and Lex-B * TF-IDF weighting resulted in the highest precision values. When inspecting higher recall values, one can observe that the Lex-C and Lex-B weighting outperformed the remaining weighting schemes with differences in precision reaching values of 10%. However, only the alignments generated with TF, TF-IDF and Lex-B * TF-IDF weighting achieved a possible recall value of 0.7 or higher, albeit at very low precision values. Another notable observation is the performance of TF-IDF based schemes. The standard TD-IDF scheme displayed performance similar to TF, thus being substantially lower than Lex-C or Lex-B. Also, the combination schemes Lex-C * TF-IDF and Lex-B * TF-IDF performed lower than their respective counterparts Lex-C and Lex-B. From this we can conclude that when applying VD-based disambiguation for a LSM, it is preferable to weight terms according to their origin and avoid the use of inverse document frequencies.

Profile Similarity

After having established the impact that different weighting techniques can have on the VD model when applied as context gathering method for a disambiguation approach, it would be interesting to see the impact of these techniques when the VD model is used for its original purpose (Qu *et al.*, 2006). Hence, in this subsection we will detail the performed experiments with the six investigated weighting schemes

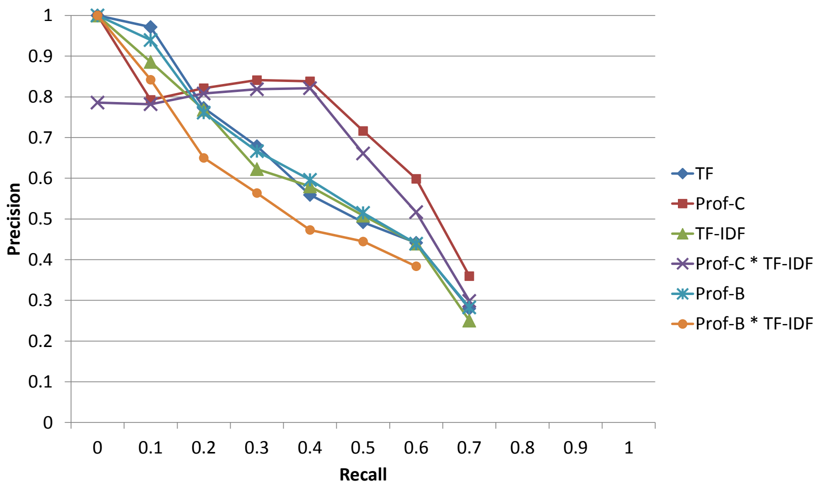


Figure 4.9: Precision versus Recall graph of the created alignments from the conference data set using the document similarities of the virtual documents.

when utilizing the virtual document model as the context gathering method for a profile similarity. All weighting schemes were used to calculate mappings on the conference data set. The measures of precision and recall were computed using the resulting alignments. The precision vs recall graph of these alignments can be seen in Figure 4.9.

From Figure 4.9 several key observations can be made. Initially, one can see that the overall two best performing schemes are Prof-C and Prof-C * TF-IDF weighting. The Prof-C * TF-IDF scheme displays a slightly worse performance than the Prof-C scheme. This indicates that the combination with TF-IDF weights not only failed to improve the term weights of the virtual documents, but rather it caused the representative strength of the VD to decrease, leading to alignments of lesser quality. The same contrast is visible when comparing Prof-B weighting with Prof-B * TF-IDF weighting.

Next, another observation can be made when contrasting the results of the TF-IDF weights with TF weights. Both schemes lead to alignments of similar quality, indicating that the combination of the inverse document frequencies to the term frequencies does not lead to the same improvements that one can observe when performing information retrieval on regular documents. Lastly, when comparing TF-IDF weighting to Prof-C and Prof-B weighting, one can see that TF-IDF weighting can at most match the performance of the other two schemes.

4.4.4 Runtime Analysis

When designing an ontology mapping framework the issue of runtime can be an important factor. This becomes an increasingly important issue when attempting to create a mapping between large ontologies, with both ontologies containing several hundred up to thousand of concepts. Adding a disambiguation procedure to a lexical similarity might cause a decrease in runtime performance, which if sufficiently significant would make in infeasible to include this approach for a large mapping task. To establish how much runtime overhead our approach generates, we executed our system on the OAEI 2013 conference data set while recording the total runtimes for the three general steps of lexical similarity measure: the retrieval of candidate senses, the disambiguation procedure and the computation of the lexical similarity. The disambiguation procedure involves the process of creating the virtual documents, document similarity computations and application of the disambiguation policy. In order to accurately establish the overhead added to the runtime of a standard lexical similarity, no word senses are discarded in the disambiguation step. As lexical similarity metric, lsm_1 was applied, though in terms of runtime there is likely to be little difference since lsm_1, lsm_2 and lsm_3 all require the computation between all pairwise combinations of senses in order to obtain their results. The recorded runtimes are presented in Table 4.4.

Computation	Sense Retrieval	Disambiguation	Lexical Similarity	Overhead
Runtime(ms)	35,272	5,632	118,900	3.65%

Table 4.4: Runtimes of the different elements of the lexical similarity on the conference dataset.

From Table 4.4 we can see that the most time consuming step of the entire similarity measure, consuming 74% of the expended computation time, is the calculation of the actual similarity values after having disambiguated all the word senses. Determining all candidate word senses for the ontology concepts, which involves several string-processing techniques such as tokenization, word-stemming and stop word removal, required 22% of the spent computational time. The creation of virtual documents and disambiguation of senses only required 3% of the computation time, meaning that the addition of this step increased the runtime by 3.65%. Given the potential performance increases of our approach, once can conclude the additional overhead introduced is negligible.

The previous comparison assumed a worse-case scenario where no senses are discarded. However, the filtering of senses can reduce the computational time for the lexical similarity by requiring fewer evaluations of the semantic similarity between senses. To see to what extent the different disambiguation policies reduce the runtime of this step, we recorded the runtimes of each policy on the conference dataset to establish possible performance gains.

From Table 4.5 we can observe that the application of the disambiguation policies can lead to significant improvements in terms of runtime. Applying the most lenient *G-Mean* policy leads to a reduction in runtime of 35 % where as the most strict policy reduces the overall runtime by 74.1 %.

Overall, we can conclude the the application of a disambiguation procedure can lead to significant improvements in runtime despite the addition of computational

Policy	Sense Retrieval	Disambiguation	Lexical Similarity	Runtime Reduction
None	35,272ms	5,632ms	118,900ms	0.0%
G-Mean	35,350ms	5,590ms	61,761ms	35.7%
A-Mean	35,780ms	5,828ms	24,847ms	58.4%
M-STD	34,229ms	5,472ms	7,244ms	70.6%
MAX	33,975ms	5,374ms	2,005ms	74.1%

Table 4.5: Runtimes of the different elements of the lexical similarity for each disambiguation policy.

overhead of the disambiguation method.

4.5 Chapter Conclusions and Future Work

We end this chapter by summarizing the results of the experiments (Subsection 4.6) and giving an outlook on future research (Subsection 4.7) based on the findings presented.

4.6 Chapter Conclusions

In this chapter we tackled *research question 1* by suggesting a method for the concept sense disambiguation of ontology concepts. The methods extends current disambiguation methods adapting information-retrieval-based techniques from contemporary profile similarities. We propose a virtual document model based on established work (Qu *et al.*, 2006) and propose a disambiguated lexical similarity capable of utilizing different disambiguation policies and methods for the computation of similarities between sets of senses.

First, we establish that the addition of our disambiguation procedure enhances the performance of lsm_1, lsm_2 and lsm_2, with the most significant improvements being observed for lsm_1 and lsm_3 (4.4.1). We further observe that the strict *MAX* disambiguation policy results in the highest measured performance of lsm_1 and lsm_3, while for lsm_2 *A-Mean* was the most effective policy. The comparison with other mapping systems using the OAEI 2011 competition and the 2013 conference dataset revealed that a otherwise modest system using our approach can result in a competitive performance when compared to established system, with our system producing higher F-measures than 50 % of the established systems and a higher precision than most systems (4.4.2). Furthermore, we investigated the possible effects on the alignment quality when applying different weighting techniques for the virtual documents. The outcome is that a weighting technique utilizing the origin of the terms within the ontology, as specified by the document model, outperforms the IR-based *TF-IDF* technique. Lastly, we establish that the addition of our proposed disambiguation approach results in an insignificant amount of computational overhead while significantly reducing the overall runtime due to the reduction of computed similarities between individual senses (4.4.4).

4.7 Future Research

We propose three directions of future research based on our findings presented in this chapter.

(1) The proposed disambiguation approach is based on existing profile similarities. While this type of similarity is fairly robust, it is still susceptible to terminological limitations and disturbances. An example of this would be the virtual documents of two concepts only containing synonymic words such that there are many equivalent but non-identical terms. While this is less of an issue for concept names, since synsets contain all synonyms for the entity they denote, this can still be an issue when comparing terms of the concept comments or synset annotations. Another example would be anomalies in the terms themselves, e.g. spelling errors or non-standard syntax for compound words. In these cases it is difficult for a profile similarity to determine an appropriate degree of overlap between terms. Future research could tackle these weaknesses through the addition of new techniques to the proposed lexical similarity. Examples of techniques which could be applied are spell-checking tools, synonym extraction techniques and soft metrics for the computation of term overlap.

(2) An alternative to tackling the weaknesses of profile similarities can be found in the combination of several disambiguation techniques. Future research could attempt to adapt other disambiguation techniques, as introduced in Subsection 4.2.1, for the purpose of concept sense disambiguation. The results of multiple disambiguation procedures would then require to be combined in a to-be-proposed aggregation strategy.

(3) The work of this chapter utilizes the lexical information available in the widely adopted WordNet dictionary. While this resource is rich with respect to the modelled entities, their relations, grammatical forms and possible labels, its additional synset annotations are typically limited to a few sentences per synset. It might be possible to achieve more accurate disambiguation results by acquiring additional information for the descriptions of each synset. New information might be gathered by querying corresponding *Wikipedia* entries by exploiting the links provided by *YAGO* or querying online search-engines such as *Google* or *Bing* and extracting terms from the search results.

Chapter 5

Anchor Profiles for Partial Alignments

This chapter is an updated and expanded version of the following publications:

1. Schadd, Frederik C. and Roos, Nico (2013). Anchor-Profiles for Ontology Mapping with Partial Alignments. *Proceedings of Twelfth Scandinavian Conference on Artificial Intelligence (SCAI 2013)*, Jaeger, Manfred, Nielsen, Thomas D. and Viappiani, Paolo ed., pp. 235−244, IOS.
2. Schadd, Frederik C. and Roos, Nico (2014a) Anchor-Profiles: Exploiting Profiles of Anchor Similarities for Ontology Mapping. *Proceedings of the 26th Belgian-Dutch Conference on Artificial Intelligence (BNAIC 2014)*, pp. 177−178.

Another type of background knowledge that can be exploited, as discussed in subsection 2.3, are partial alignments. Given the two input ontologies O_1 and O_2, a partial alignment PA specifies an alignment that is incomplete with respect to the entities in O_1 and O_2. Essentially, if a domain expert were to be presented with PA, he would not be satisfied with PA until a series of correspondences are added to the alignment. The main goal in this scenario is to identify the additional correspondences which the domain expert would add to PA.

This chapter addresses the second research question by proposing a profile-based method of utilizing the anchors of a partial alignment. We generalize the notion of a profile such that it expresses the affinity to certain objects. In a classic profile approach, the objects denote natural language terms and the affinity to a term is expressed by how often the term occurs appears in the vicinity of the given concept. We propose an alteration of that interpretation, such that a profile expresses the concept's affinity to a series of given anchors. The core intuition behind this approach is that concepts which denote the same entity are more likely to exhibit the same levels of affinity to the given anchors. We evaluate our approach on the OAEI benchmark dataset. Particularly, we investigate the effects of partial alignment sizes and correctness and compare the performance of the approach to contemporary mapping systems.

The remainder of this chapter is structured as follows. Section 5.1 discusses work that is related to mapping with partial alignments. Section 5.2 details our proposed approach. Section 5.3 presents and discusses the results of the performed experiments while Section 5.4 presents the conclusion of this chapter.

5.1 Related Work

Several works exist that have described approaches which reuse previously generated alignments. The principle behind these approaches has initially been suggested by Rahm and Bernstein (2001). Here, the focus lies on finding auxiliary ontologies which are already mapped to the target ontology. This has the intention that, by selecting the auxiliary ontology according to a specific criteria, the remaining mapping problem between the source and auxiliary ontology might be easier to solve than the original problem. Subsequent works have expanded this idea to deriving mappings when both input ontologies have an existing alignment to an auxiliary ontology.

COMA++ employs several strategies with respect to exploiting pre-existing alignments (Aumueller *et al.*, 2005). Most prominently, the system can explore alignment paths of variable lengths between multiple ontologies, which are obtained from a corpus, in order to derive its mappings. It is also possible to explore ontologies from the semantic web for this purpose (Sabou *et al.*, 2008). The resulting mapping derivations of multiple alignment paths can be combined to form a more reliable mapping.

While the previously mentioned approaches utilized complete mappings involving auxiliary ontologies, there has been some research into approaches that exploit partial alignments that exist between the source and target ontologies. These alignments can either be user generated, by for instance using the PROMPT tool (Noy and Musen, 2000), or automatically generated from a different system.

The most prominent approach is the Anchor-PROMPT (Noy and Musen, 2001) algorithm. Here, possible paths between anchors are iteratively explored in parallel in both ontologies, while the encountered concept combinations are registered. The intuition is that concept pairs which have been encountered regularly during the exploration phase are more likely to correspond with each other.

The Anchor-Flood algorithm also features a type of iterative exploration by exploiting anchors (Seddiqui and Aono, 2009). This approach selects a main anchor and iteratively expands the explored neighbourhood of this anchor. At each iteration, a matching step is invoked which compares the concepts in this neighbourhood and updates the alignment if new correspondences are found. A similar procedure can be seen in the LogMap system (Jiménez-Ruiz and Cuenca Grau, 2011; Jiménez-Ruiz *et al.*, 2012b). This system alternates between an anchor-based discovery step and a mapping repair step in order to compute a full mapping.

5.2 Anchor Profiles

A profile similarity gathers context information of ontology concepts and compares these context collections by parsing them into a vector space and comparing the resulting vectors. This context information can consist of data from the concept description and the descriptions of related concepts (Qu *et al.*, 2006; Mao *et al.*, 2007). The intuition behind this approach is that concepts can be considered similar if they have similar context information. More generally, a profile can be considered as a vector generated from data which describes a concept, hence two concepts are similar if their profiles can be considered similar.

When mapping two ontologies for which a partial alignment is provided by a domain expert, new opportunities arise when selecting similarity measures for a mapping system. Instead of using description information as the basis for a profile, we suggest utilizing the correspondences of the given partial alignment, also referred to as anchors, as basis for a new kind of profile similarity. Here, since the anchors are assumed to be correct, the main intuition is that two concepts can be considered similar if they exhibit a comparable degree of similarity towards the given anchors. We will illustrate this intuition with an example, depicted in Figure 5.1.

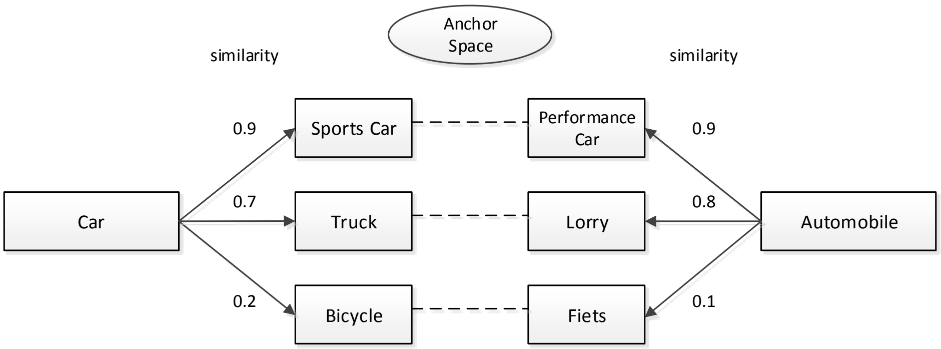


Figure 5.1: Two equivalent concepts being compared to a series of anchors.

Figure 5.1 depicts two classes, *Car* and *Automobile*, being compared to three given anchors, <*Sports Car, Performance Car*>, <*Truck, Lorry*> and <*Bike, Fiets*>. When comparing *Car* and *Automobile* to a concept of the anchor <*Sports Car, Performance Car*> it is reasonable to expect that both comparisons would result in a high value since they are highly related. On the contrary, comparing *Car* and *Automobile* to the anchor <*Bike, Fiets*> is likely to result in a lower value, since a bike is not closely semantically related to the concept of a car. However, since *Car* and *Automobile* are semantically equivalent, one can expect the resulting values of the comparison to be equally low.

In order to compare concepts with anchors we need to define a metric capable of doing so. Given two ontologies O_1 and O_2, and given an anchor $A_x[C^1, C^2]$ containing a correspondence between the concepts C^1 and C^2 originating from O_1 and

O_2 respectively, and given a concept similarity $sim'(E,F) \in [0,1]$ which expresses the similarity between two concepts, we define an anchor similarity $sim_A(C,A_x)$ between an arbitrary concept C and A_x as:

$$sim_A(C,A_x) = \begin{cases} sim'(C,C^2) & \text{if } C \in O_1 \\ sim'(C,C^1) & \text{if } C \in O_2 \end{cases} \tag{5.1}$$

Note that C is compared to the concept in the anchor which originates from the other ontology. If one were to compare C to the anchor concept from the same ontology, sim' would be reduced to a structural similarity, similar to a taxonomy distance, making the distinction between classes that are related equivalently close to a given anchor prohibitively difficult. We will empirically demonstrate the effectiveness of this anchor similarity, opposed to comparing C to the anchor concept originating from the same ontology, in subsection 5.3.3. From equation 5.1 follows that two concepts C and D can be considered similar if $sim_A(C,A_x)$ and $sim_A(D,A_x)$ are similar. Given that a partial alignment most likely contains multiple correspondences, this intuition needs to be expanded for a series of anchors. This brings us back to the generalized idea of a profile, such that we can use the anchor similarities sim_A between a concept C and all anchors as the basis of a profile, referred to as anchor-profile.

Algorithm 5.1 Anchor-Profile Similarity

```
Anchor-Profile($E_1$,$E_2$,$PA$,$sim_A$, $sim_P$)

for all $e \in E_1 \cup E_2$ do
    $Profile(e) \leftarrow initVector(|PA|)$
    for all $i = 1$ to $|PA|$ do
        $A_i \leftarrow PA[i]$
        $Profile(e)(i) \leftarrow sim_A(e, A_i)$
    end for
end for
$M \leftarrow initMatrix(|E_1|, |E_2|)$
for $i = 1$ to $|E_1|$ do
    for $j = 1$ to $|E_2|$ do
        $e_1 \leftarrow E_1[i]$
        $e_2 \leftarrow E_2[j]$
        $M[i,j] \leftarrow sim_P(Profile(e_1), Profile(e_2))$
    end for
end for
return $M$
```

Formally, let us define *Profile*(e) as the profile of the entity e. Also, let us define sim_P as a similarity metric capable of comparing two profiles, as introduced in subsection 3.1.2. Given two sets of entities E_1 and E_2, belonging to the ontologies O_1 and O_2 respectively, a partial alignment $PA = \{A_1, A_2, \ldots, A_n\}$ consisting of n anchors, an anchor similarity sim_A and a similarity measure sim_P, we compute the matrix M of anchor-profiles similarities between E_1 and E_2 as defined in Algorithm 5.1.

Figure 5.2 visualizes our anchor-profile similarity, as defined in Algorithm 5.1.

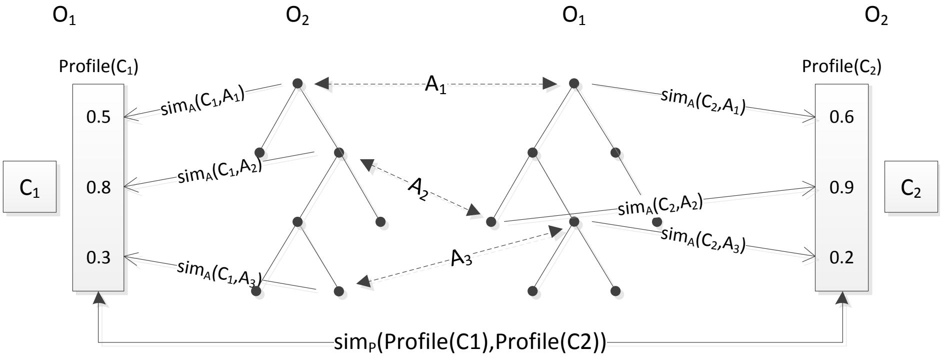


Figure 5.2: Visualization of an anchor profile similarity.

The example in Figure 5.2 shows two ontologies, O_1 and O_2, and three anchors A_1, A_2 and A_3. Two concepts C_1 and C_2, originating from O_1 and O_2 respectively, are compared using their respective anchor-profiles *Profile*(C_1) and *Profile*(C_2). The profile vectors are compared using the similarity sim_P. While there exist various similarity measures for vectors, for this research the well-known cosine-similarity (Pang-Ning *et al.*, 2005) has been applied as sim_P.

Since the main intuition of this approach is that corresponding concepts should exhibit a comparable degree of similarity towards the given anchors, it is necessary to choose sim' such that this metric is robust under a wide variety of circumstances. Since every single metric has potential weaknesses (Shvaiko and Euzenat, 2005), it is preferable to aggregate different metrics in order to overcome these. To realise this, sim' utilizes the aggregate of all similarities from the *MaasMatch* system (Schadd and Roos, 2012b). Figure 5.3 displays the configuration of the evaluated mapping system. Here, two distinct similarity matrices are computed, being the similarities of the anchor-profiles and an aggregate of other metrics. This second matrix is necessary for the eventuality where the system has to differentiate correspondences that all contain anchor-profiles which closely resemble null vectors, which occurs when a concept displays no similarity to any of the given anchors. This can occur when a given ontology has a considerable concept diversity and the given anchors do not adequately cover the concept taxonomy. The aggregation of these two matrices is then used to extract the output alignment A.

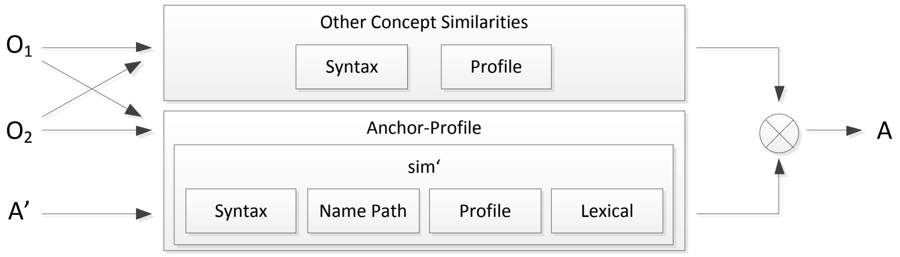


Figure 5.3: Overview of the tested mapping system.

5.3 Experiments

To evaluate the performance of our approach, we will use the measures of adapted Precision P^*, adapted Recall R^* and adapted F-Measure F^*, as introduced in Section 2.3. Furthermore, we will also compute the standard measures of Precision, Recall, and F-Measure when it is necessary to establish the overall quality of the resulting alignments. This section is structured into the following subsections, which individually either establish the performance of our approach or investigate interesting properties:

- Subsection 5.3.1 establishes the overall performance of the approach with an evaluation on the OAEI benchmark dataset.
- Subsection 5.3.2 analyses the benchmark results in more detail by analysing the performance over the different tasks of the dataset.
- We evaluate the intuition behind sim_A by comparing its performance to an alternative anchor-similarity in subsection 5.3.3.
- In subsection 5.3.4 we investigate to what extend incorrect anchors influence the performance of our approach.
- Subsection 5.3.5 compares the quality of the produced correspondences to other contemporary matching systems.

In order to evaluate the performance of a mapping approach which exploits partial alignments, it is necessary to have access to a dataset which not only contains appropriate mapping tasks and their reference alignments, but also partial alignments that can be used as input. However, within the boundaries of the OAEI competition, which allows a comparison with other frameworks, there does not exist a dataset which also supplies partial alignments as additional input with which a recent evaluation has taken place. When a dataset does not contain partial alignments, it is possible to generate these by drawing correspondences from the reference alignment at random. In order to account for the random variation introduced by the generated partial alignments, it becomes necessary to repeatedly evaluate the dataset using many generated partial alignments for each mapping task. The values of precision, recall and F-measure can then be aggregated using the arithmetic mean.

Next to establishing the mean performance of a system, it is also interesting to see how stable its performance is. Traditionally, this is expressed via the standard deviation. However, given that in this domain the measurements origin from different tasks of differing complexity, this introduces a problem. Given the presence of tasks of varying complexity that can occur in a dataset, it is to be expected that the mean performances of the repeated evaluations differ for each task. Thus, in order to combine the standard deviations of the different tasks, a statistical measure is needed that takes this into account. To do this we propose using the pooled standard deviation of the different measures (Dodge, 2008; Killeen, 2005).

Given k samples, the different sample sizes $n_1, n_2, \ldots, n_k$ and sample variances $s_1^2, s_2^2, \ldots, s_k^2$, the pooled standard deviation of the collection of samples can be calculated as follows:

$$s' = \sqrt{\frac{(n_1 - 1) \times s_1^2 + (n_2 - 1) \times s_2^2 + \cdots + (n_k - 1) \times s_k^2}{n_1 + n_2 + \cdots + n_k - k}} \tag{5.2}$$

In this domain, the repeated evaluation of a single track using randomly generated partial alignments can be viewed as a sample, such that the pooled standard deviation expresses how much the results deviate across all tracks. For the remainder of this chapter, we will refer to the pooled standard deviation of P^*, R^* and F^* as s'_{P^*}, s'_{R^*} and s'_{F^*} respectively.

5.3.1 Evaluation

To evaluate an anchor-profile approach, an ontology mapping system incorporating the proposed similarity has been evaluated on the *benchmark-biblio* dataset originating from the 2012 Ontology Alignment Evaluation Initiative (Aguirre *et al.*, 2012). This synthetic dataset consists of tasks where each task tests a certain limiting aspect of the mapping process, for instance by distorting or removing certain features of an ontology like concept names, comments or properties. Since this dataset does not contain partial alignments that can be used as input, they were randomly generated from the reference alignments. In order to evaluate what impact the size of the partial alignment can have on the mapping process, we evaluated our approach over a spectrum of partial alignment recall values $[0.1, 0.2, \ldots, 0.9]$. Thus, as an example, a partial alignment recall of 0.2 indicates that PA was randomly generated from the reference R such that PA has a recall of 0.2. In order to mitigate the variance introduced through the random generation of PA, each recall level has been evaluated 100 times where each evaluation contained a new set of randomly generated partial alignments. For each evaluation, the adapted measures of precision, recall and F-measure, P^*, R^* and F^* respectively, were computed and aggregated. Table 5.1 displays the aggregated results of the evaluation.

From Table 5.1, several interesting results and trends can be seen. First, we can see that overall for all PA recall levels the system resulted in an adapted precision

PA Recall	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
P^*	0.760	0.769	0.779	0.786	0.801	0.817	0.835	0.855	0.866
R^*	0.632	0.641	0.649	0.656	0.663	0.674	0.685	0.702	0.745
F^*	0.668	0.678	0.686	0.693	0.701	0.713	0.726	0.743	0.780
s'_{P^*}	0.094	0.099	0.107	0.112	0.125	0.139	0.155	0.180	0.219
s'_{R^*}	0.049	0.068	0.083	0.092	0.102	0.117	0.133	0.158	0.215
s'_{F^*}	0.038	0.053	0.066	0.074	0.083	0.098	0.115	0.142	0.199

Table 5.1: Results of the evaluations on the benchmark-biblio dataset using different recall requirements for the randomly generated partial alignments. For each recall requirement, 100 evaluations were performed and aggregated.

in the interval $[0.76, 0.87]$, adapted recall in the interval $[0.63, 0.75]$ and adapted F-measure in the interval $[0.66, 0.78]$. Thus, for every *PA* recall level the approach resulted in high precision and moderately high recall measure. Furthermore, we can observe that as the recall of *PA* increases, the adapted precision, recall and F-measure of A increase as well. This increase is fairly consistent over all *PA* recall levels, indicating that a larger amount of anchors improves the representative strength of the computed anchor profiles.

Inspecting s'_{P^*}, s'_{R^*} and s'_{F^*} reveals each measure shows a similar trend. For each measure, an increase of the recall level of *PA* also yields an increase of the pooled standard deviation, with the resulting alignments at *PA* recall level of 0.1 being fairly stable, while a moderate variance can be observed at a *PA* recall level of 0.9. This trend is to be expected since any variation in A will have a larger impact on P^*, R^* and F^* if *PA* has a significant size.

5.3.2 Performance Track Breakdown

Having established the overall performance of the proposed approach for different size levels of *PA*, it would be interesting to inspect the performance over the different tasks of the dataset. The task groups reflect different kinds of alterations in the target ontology and have been grouped as follows:

101 A baseline task where the complete ontology is matched against itself. Allows for the identification of any fundamental flaws in a mapping system.

201-202 Concept names have been removed. Task 202 has concept description removed in addition.

221-228 Testing the separate removal or alteration of instances, the class hierarchy, restrictions or properties.

232-247 Testing all possible combinations of removing or altering the instances, hierarchy, restrictions or properties.

248-253 Similar to 221-228, this group tests the removal or alteration of instances, the class hierarchy, restrictions or properties. However, concept names and descriptions have been removed in addition.

254-266 Similar to 232-247, this group tests all possible combinations of removing or altering the instances, hierarchy, restrictions or properties. However, concept names and descriptions have been removed in addition.

Figures 5.4, 5.5 and 5.6 show adapted precision, recall and F-measure over several task groups when using different *PA* size levels, ranging from 0.1 to 0.9 in intervals of 0.1.

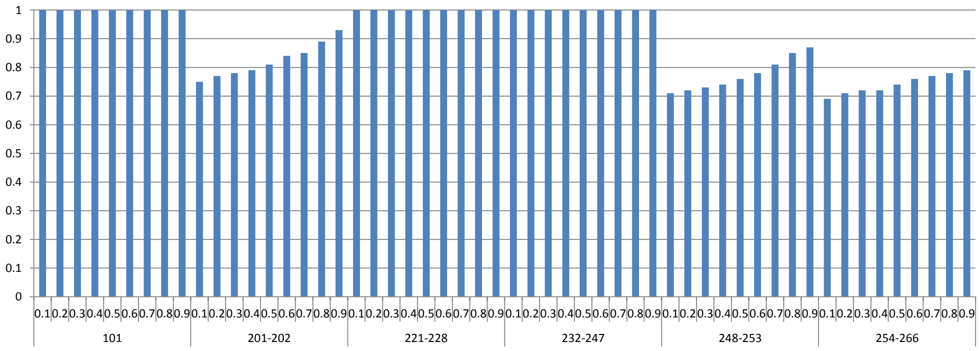


Figure 5.4: Corrected precision of the proposed approach for the different task groups of the benchmark dataset. Each group contrasts the performance of different partial alignment recall levels.

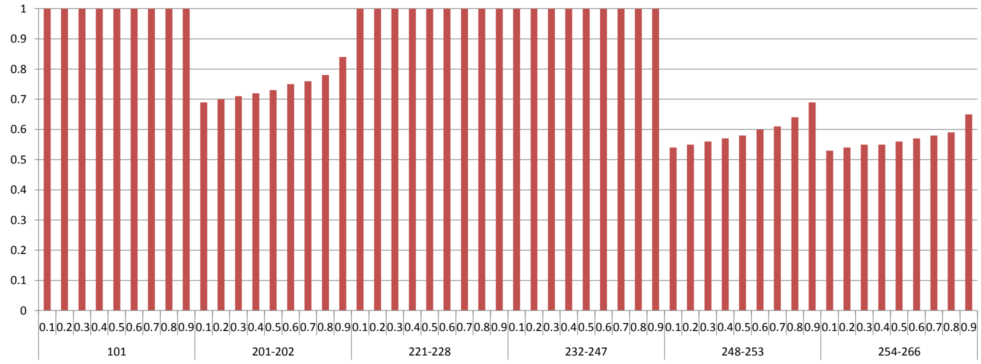


Figure 5.5: Corrected recall of the proposed approach for the different task groups of the benchmark dataset. Each group contrasts the performance of different partial alignment recall levels.

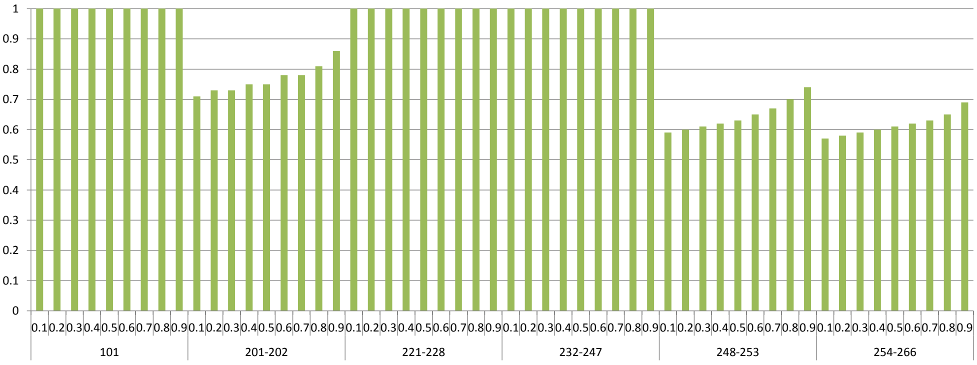


Figure 5.6: Corrected F-measure of the proposed approach for the different task groups of the benchmark dataset. Each group contrasts the performance of different partial alignment recall levels.

From this evaluation we can observe two clear trends. Firstly, the performance in terms of precision, recall and F-measure is positively correlated with the size of PA. We can see most prominently in track groups 201-202 and 248-253 that the performance improvements surpass linear growth, as we can observe more substantial improvements for larger PA sizes. Secondly, we can see a divided in performance when contrasting the groups 101, 221-228 and 232-247 against the groups 201-202, 248-253 and 254-266, with the former being matched with perfect results. This can be explained by the setup of the mapping system, specifically the similarities being employed in sim'. These similarities are predominantly syntactic and structural, such that the track groups where these traits are not altered display a justifiably high performance. A more varied set of similarities should improve the performance on the remaining track groups.

5.3.3 Alternate Profile Creation

The basis of the Anchor Profile approach is the comparison of ontology concepts to anchors, which is achieved via sim_A as described in equation 5.1. Here, an ontology concept C is compared to an anchor A by retrieving the concept from A which does not originate from the same ontology as C. While in section 5.2 we elaborated the intuition behind this, we will empirically demonstrate the correctness of this approach by comparing sim_A to an anchor similarity which compares C to the anchor-concept which originates from the same ontology as C. To achieve this, given two ontologies O_1 and O_2, and given an anchor $A_x[C^1, C^2]$ containing a correspondence between the concepts C^1 and C^2 originating from O_1 and O_2 respectively, we define a new anchor similarity sim^*_A as follows:

$$sim^*_A(C, A_x) = \begin{cases} sim'(C, C^2) & \text{if } C \in O_2 \\ sim'(C, C^1) & \text{if } C \in O_1 \end{cases} \tag{5.3}$$

Having defined sim_A^*, an evaluation was performed similar to the evaluation in subsection 5.3.1, however with sim_A^* substituted for sim_A. Figure 5.7 compares the resulting adapted precision values of these evaluations.

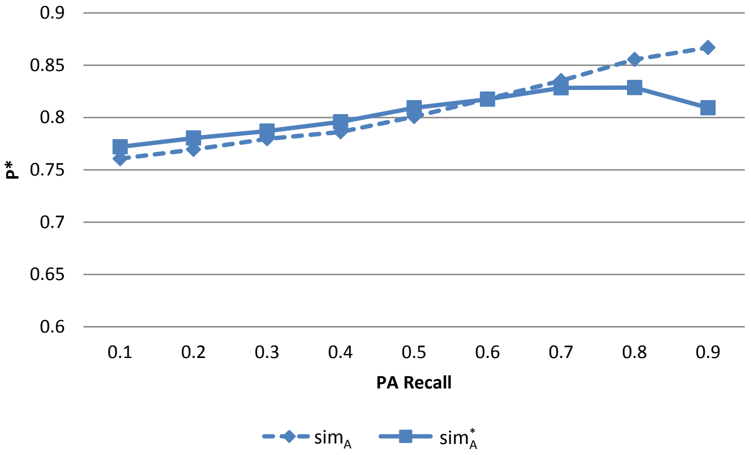


Figure 5.7: Adapted precision of the anchor profile approach using sim_A and sim_A^* as anchor similarities.

From Figure 5.7 we can see that sim_A^* produces slightly higher precision values for low PA recall levels, with an adapted precision difference of approximately 0.01 for every PA recall value up to 0.5. However, for higher PA recall values, where the anchor profile has a higher dimensionality, we can observe a strong increase in precision when using sim_A. On the contrary, the adapted precision when using sim_A^* stagnates and even drops for these higher PA recall values, leading maximum difference of adapted precision of approximately 0.06. We can conclude that, when using sim_A^*, that the resulting adapted precision is not positively correlated with the recall of PA, unlike sim_A.

Next, the resulting adapted recall values for the evaluated PA recall levels are compared, when applying sim_A and sim_A^* , are compared in Figure 5.8.

Unlike the comparison of the adapted precision, in Figure 5.8 we can observe a more straight-forward result. While for both sim_A and sim_A^* the adapted recall is positively correlated with the recall of PA, applying sim_A resulted in significantly higher adapted recall values for all PA recall values, with the difference ranging from 0.06 up to values of 0.08.

Finally, the resulting adapted F-measures, indicative of the overall performance of both anchor similarities, are compared in Figure 5.9.

In Figure 5.9 we can see a similar trend is in Figure 5.8, with the adapted F-measure being significantly higher at all PA recall values, though with the difference slightly less pronounced due to sim_A producing slightly lower adapted precision values, as seen in Figure 5.7. However, we can observe a minimum adapted F-measure difference of 0.032, at a PA recall of 0.1 and a maximum adapted F-measure

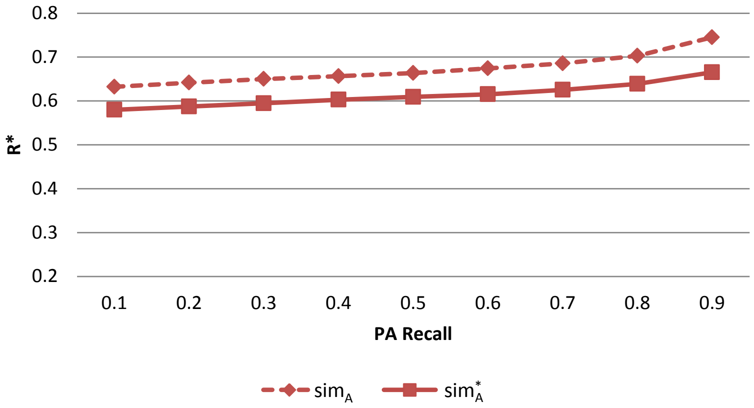


Figure 5.8: Adapted recall of the anchor profile approach using sim_A and sim_A^* as anchor similarities.

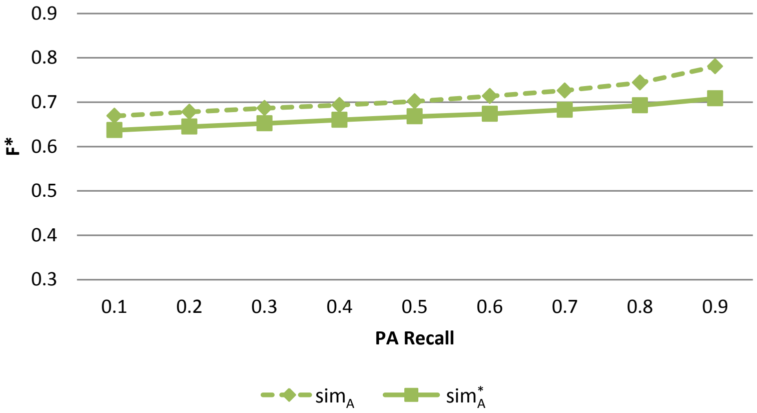


Figure 5.9: Adapted F-measure of the anchor profile approach using sim_A and sim_A^* as anchor similarities.

difference of 0.072 at PA recall of 0.9.

Overall, we can conclude that by comparing ontology concepts to the anchor concept originating from the opposing ontology results in a superior quality of computed alignments for all PA recall values.

5.3.4 Influence of Deteriorating PA Precision

As previously stated, the general assumption of an approach which utilizes partial alignments is that the correspondences within the partial alignment can be assumed to be correct. This assumption is based on the fact that partial alignments are generated by a domain expert or by a specialized pre-processing technique. However, it can be possible that a domain expert can make an error, or that the specialized pre-processing technique does not produce correct correspondences with 100% certainty, in which case the performance of the Anchor Profile approach, or any other approach which utilizes partial alignments, might suffer. In this subsection we will investigate to what extent the performance of the Anchor Profile approach is influence in the eventuality that this assumption is wrong and that the partial alignment contains incorrect correspondences. Formally, given a partial alignment PA and a reference alignment R, we will investigate the situation in which $|PA \cap \overline{R}| \neq \emptyset$. From this follows that $P(PA, R) < 1$. We will systematically evaluate the OAEI 2012 benchmark dataset with PA recall levels ranging from 0.1 to 0.9, similar to subsection 5.3.1. However, for each PA recall level we will evaluate a series of PA precision levels, ranging from 0.1 to 0.7. This will provide an indication of performance degradation for the relative spectrum of PA sizes. Table 5.2, 5.3 and 5.4 display the adapted precision, recal and F-measure, respectively, of this evaluation.

	$R(PA,R)$								
$P(PA,R)$	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
1	0.760	0.769	0.779	0.786	0.801	0.817	0.835	0.855	0.866
0.9	0.766	0.763	0.771	0.783	0.775	0.787	0.786	0.784	0.592
0.8	0.755	0.755	0.749	0.758	0.752	0.748	0.735	0.498	0.769
0.7	0.759	0.741	0.742	0.727	0.722	0.704	0.510	0.136	0.690
0.6	0.746	0.728	0.713	0.703	0.696	0.456	0.752	0.737	0.553
0.5	0.732	0.714	0.695	0.684	0.324	0.724	0.701	0.614	0.379
0.4	0.726	0.683	0.675	0.345	0.688	0.648	0.554	0.408	0.187
0.3	0.711	0.666	0.335	0.666	0.569	0.433	0.279	0.168	0.054

Table 5.2: Adapted precision $P^*(A, R)$ of the Anchor Profile approach for varying recall and precision levels of the input partial alignment.

First, we can see that the performance of the anchor profile approach is negatively affected upon decreasing $P(PA, R)$. While this is an expected result, it is interesting to see to what extent this decrease occurs for the different values of $R(PA, R)$. For the smallest value of $R(PA, R)$, we can observe a gradual decrease in F-measure, mostly caused by a decrease in the recall of the result alignments. However, when increasing $R(PA, R)$ we can observe that the decrease in adapted F-measure occurs more quickly and steeply. For very large values of $R(PA, R)$ we can even observe a non-gradual decline precision and recall values. This stems from the nature of the input partial alignment. When given a partial alignment which already contains a large portion of the reference alignment, only a few correct correspondences remain to be discovered. The actual amount of remaining correspondences also varies due to the random addition of incorrect correspondences, since any concept in PA is not matched again due to the implicit assumption of the correctness of PA.

Next, while one would initially presume that despite the presence of incorrect

	$R(PA,R)$								
$P(PA,R)$	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
1	0.632	0.641	0.649	0.656	0.663	0.674	0.685	0.702	0.745
0.9	0.634	0.603	0.604	0.608	0.545	0.537	0.456	0.388	0.274
0.8	0.598	0.562	0.517	0.508	0.436	0.353	0.250	0.112	0.493
0.7	0.598	0.523	0.476	0.382	0.300	0.180	0.084	0.068	0.379
0.6	0.562	0.457	0.365	0.271	0.157	0.056	0.306	0.286	0.260
0.5	0.529	0.391	0.270	0.157	0.028	0.233	0.201	0.175	0.156
0.4	0.465	0.282	0.146	0.028	0.168	0.135	0.108	0.087	0.069
0.3	0.404	0.164	0.023	0.130	0.081	0.056	0.038	0.028	0.018

Table 5.3: Adapted recall $R^*(A,R)$ of the Anchor Profile approach for varying recall and precision levels of the input partial alignment.

	$R(PA,R)$								
$P(PA,R)$	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
1	0.668	0.678	0.686	0.693	0.701	0.713	0.726	0.743	0.780
0.9	0.669	0.651	0.654	0.661	0.619	0.617	0.557	0.499	0.363
0.8	0.645	0.622	0.591	0.587	0.532	0.461	0.357	0.178	0.579
0.7	0.645	0.593	0.560	0.482	0.407	0.274	0.140	0.086	0.471
0.6	0.619	0.542	0.465	0.375	0.245	0.097	0.415	0.393	0.342
0.5	0.595	0.487	0.373	0.243	0.052	0.335	0.297	0.261	0.215
0.4	0.547	0.384	0.229	0.051	0.257	0.213	0.173	0.139	0.100
0.3	0.497	0.251	0.042	0.207	0.136	0.096	0.066	0.048	0.027

Table 5.4: Adapted F-measure $F^*(A,R)$ of the Anchor Profile approach for varying recall and precision levels of the input partial alignment.

correspondences in PA, the quality of the output alignments would remain constant if the ratio between correct and incorrect correspondences in PA remains constant as well. However, the experiment reveals that this is rarely the case. For $P(PA,R)$ values of 0.9, we can observe that the average F-measure is mostly constant, with the exceptions occurring at high $R(PA,R)$ values. Intriguingly, this constant performance is achieved due the rising precision compensating for the decreasing recall. For $P(PA,R)$ values of 0.8 and lower we can see that an increase of $R(PA,R)$ actually leads to a detriment in average F-measure. Since fraction of incorrect correspondences remains constant, we can conclude that the absolute amount of incorrect correspondences also is a factor for the observed performance.

From this experiment, we can conclude that the precision of the input partial alignments plays an important factor for the performance of the Anchor profile approach, and likely also for any other approach which utilizes partial alignments.

It follows that any future deployment of this techniques requires a preprocessing step which would ensure that the precision of the partial alignments lies as closely to 1 as possible. The evaluation shows that, if such a pre-processing step would have the disadvantage of reducing the recall of *PA*, the overall performance would still be impacted positively.

5.3.5 Comparison with other Frameworks

Next to establishing the overall performance on the benchmark dataset, it is also important to provide some context to that performance. To do this, we will compare the performance of the Anchor-Profile approach with the top 8 frameworks out of 18 frameworks that participated in the OAEI 2012 competition (Aguirre *et al.*, 2012) in Table 5.5. Unfortunately, none of the OAEI evaluations contained a task which also provided partial alignments, however a comparison with state-of-the-art systems which tackled the same task without a partial alignment can still be a useful performance indication. For this comparison, both the smallest and largest evaluated *PA* size levels were used.

System	Precision	Recall	F-Measure
MapSSS	0.99	0.77	0.87
YAM++	0.98	0.72	0.83
Anchor-Profile (0.9)	0.866*(0.998)	0.745*(0.967)	0.78*(0.982)
AROMA	0.98	0.64	0.77
WeSeE	0.99	0.53	0.69
AUTOMSv2	0.97	0.54	0.69
Hertuda	0.9	0.54	0.68
Anchor-Profile (0.1)	0.760*(0.88)	0.632*(0.623)	0.668*(0.691)
HotMatch	0.96	0.5	0.66
Optima	0.89	0.49	0.63

Table 5.5: Comparison of the Anchor-Profile approach, using two different PA thresholds, with the 8 best performing frameworks from the OAEI 2012 competition. An asterisk indicates the value has been adapted with respect to PA, while the values inside the brackets indicate the respective measure over the entire alignment.

The results of Table 5.5 indicate that the quality of correspondences produced by our approach is in line with the top ontology mapping frameworks in the field. In fact, when including *PA* in the evaluation metrics, the anchor-profile approach outperforms these frameworks given a large enough recall level[1] of *PA*. Using partial alignments with a recall of 0.1 resulted in an F-measure similar to the HotMatch framework, ranking at 8th place in this comparison. A *PA* recall level of 0.9 resulted in a sufficiently high F-measure to rank 3rd among the top ranking systems. With regards to precision and recall, our system differentiates itself from other frameworks

[1] The results of the experiments indicate that a recall level of 0.5 would suffice.

by having a comparatively lower precision and higher recall. This indicates that our approach is capable of identifying correspondences which other system cannot, while further measures must be implemented to differentiate between correspondences that have similar anchor profiles.

5.4 Chapter Conclusions and Future Work

We end this chapter by summarizing the results of the experiments (Subsection 5.4.1) and giving an outlook on future research (Subsection 5.4.2) based on the findings presented.

5.4.1 Chapter Conclusions

In this chapter we tackled *research question 2* by proposing the Anchor-Profile technique. We generalized the notion of a profile as an expression of affinity towards a series of objects, such that two equivalent concepts are more likely to exhibit the same degrees of affinity. Our Anchor-Profile approach uses the anchors of a partial alignment for the measurement of affinities. We proposed a method capable of using established similarities in order to create the profiles of each concept.

First, we demonstrated the performance of our approach with an evaluation on the OAEI benchmark dataset (5.3.1). For this dataset, we generated the partial alignment by randomly sampling from the reference alignment and aggregated the results of multiple executions using numerous sampled partial alignments. In this experiment we established that the matching performance is positively influenced by the recall of the provided partial alignment PA by performing a full evaluation of the benchmark dataset for different recall values of the sampled partial alignments, spanning the interval of $[0.1, 0.9]$ using increments of 0.1. We also noted that the pooled standard-deviations s'_{P*}, s'_{R*} and s'_{F*} increase for higher partial alignment recalls. This is explained by the fact that the variations of P^*, R^* and F^* will be more pronounced if PA has a significant size. Second, we analysed the performance of our approach in more detail by inspecting the results of the benchmark evaluation for the different provided task groups (5.3.2). We concluded that the choice of sim' reflects on the performance of the Anchor-Profile approach, since in order to adequately compare a concept with an anchor it is necessary that both contain the meta-information that is relevant to *sim*'.

In a further experiment we validated the intuition behind our choice for sim_A by comparing its performance against an alternative sim^*_A which compares concepts to the anchor-concept of the same ontology (5.3.3). Next, we investigated the possible effect incorrect anchors can have on the alignment quality. For this, we performed a systematic evaluation on the benchmark dataset by sampling the partial alignments according to a series of combinations between specified $P(PA, R)$ and $R(PA, R)$ values (5.3.4). From this evaluation we concluded that the correctness of the anchor can indeed have a significant impact on the alignment quality, particularly for larger partial alignments. Lastly, we provided a context for the performance of our approach by comparing the achieved results to the performance of the top 8 systems that were evaluated on the same dataset in the OAEI competition (5.3.5). We observed that the alignment quality of our approach is comparable with the quality of state-of-the-art system, surpassing the performance of most systems for larger sizes of *PA*.

5.4.2 Future Research

We propose two directions of future research based on our findings presented in this chapter.

(1) In Subsection 5.3.2 we observed that the performance over the different categories of matching tasks, categorized by the types and combinations of different kind of heterogeneities, is influenced by the choice of sim'. To improve the robustness of our approach we selected sim' as a combination of different types of similarities. This selection has been shown to be susceptible to strong terminological disturbances, as seen in the performance over the matching tasks 248-253 and 254-266. Further research should evaluate the effects of different selections of sim' and whether a higher performance can be achieved by utilizing techniques which do no exploit terminological meta-information.

(2) Other mapping systems can utilize anchor-based techniques by generating a set of anchors during the matching process. Future research could investigate the performance of our technique when utilized in a similar manner. Particularly, the research should be focused on the deployed anchor-generation techniques, as we have shown in Subsection 5.3.4 that both the size and quality of the anchors can significant impacts on the performance.

Chapter 6

Anchor Evaluation using Feature Selection

This chapter is an updated version of the following publication:

1. Schadd, Frederik C. and Roos, Nico (2014c). A Feature Selection Approach for Anchor Evaluation in Ontology Mapping. *Knowledge Engineering and the Semantic Web*, Klinov, Pavel and Mouromtsev, Dmitry ed., pp. 160−174, Springer, *Top 5 Research Paper Award Winner.*

In the previous chapter we have introduced a method for mapping ontologies using partial alignments. Further, we have established that the performance of this method depends not only on the size of the supplied partial alignment but also on its correctness. This implies that the performance of partial-alignment-based matchers will also be affected by these qualities.

The third research question has been formulated as a response to the obtained results of answering the second research question, particularly with respect to the influence of the quality of the provided partial alignments. This chapter aims to answer the third research question by proposing a novel method facilitating the evaluation of the provided anchors. The results of this evaluation can be used to apply a filtering method in order to ensure the quality of the provided anchors. To achieve this, our proposed method is aimed at exploiting the set of correct correspondences which can be reliably generated with a pairwise similarity metric. We compare how well a provided anchor aligns with a single generated correspondence by proposing a dissonance measure. Further, we quantify how well an anchor aligns with multiple correspondences by formulating this evaluation as a feature selection task, originating from the field of machine learning. We evaluate our method using the OAEI conference dataset when applying numerous configurations.

The remainder of this chapter is structured as follows. Section 6.1 formalizes the task of evaluating and filtering anchors when matching with partial alignments. Section 6.2 details our approach for the subtask of evaluating anchors. Section 6.3 presents and discusses the results of the performed experiments. The conclusions of this chapter and a discussion of future research are presented in Section 6.4.

6.1 Anchor Filtering

While the correspondences originating from a partial alignment, referred to as anchors, can be assumed to be correct, this is not always the case. In case of a generated partial alignment, there is no guarantee that the used approach has a precision of 100% for every mapping task. If the partial alignment is made by a domain expert, it can always occur that the expert makes a mistake. The presence of incorrect anchors can degrade the quality of the computed correspondences, with the degradation of quality being correlated to the quantity of incorrect anchors. In order to ensure that a mapping approach that utilizes partial alignments performs as designed, it becomes necessary to perform a pre-processing step that ensures that the provided anchors are of sufficient quality.

The procedure of pre-processing partial alignments can be described by two key steps: *anchor evaluation* and the application of a *filtering policy.* Given two ontologies O_1 and O_2, and a partial alignment PA consisting of n anchors $\{a_1, a_2, \ldots a_n\}$, the *anchor evaluation* step produces a set of n scores $S = \{s_1, s_2, \ldots, s_n\}$, with each score s_x indicating the quality of its anchor c_x. The *filtering* step uses these scores to discard any anchor which does not satisfy a given policy, creating a new partial alignment PA', such that $PA' \subseteq PA$. The entire process is illustrated in Figure 6.1.

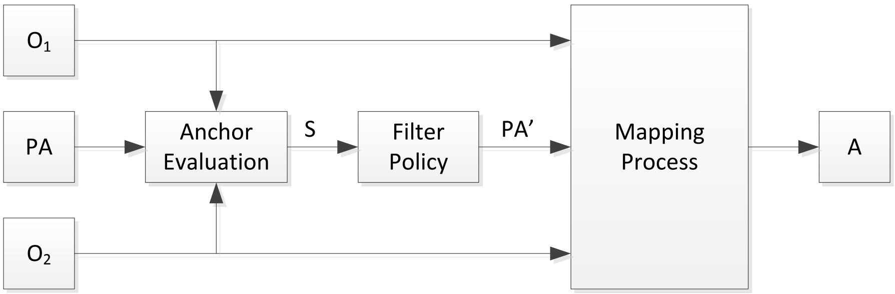


Figure 6.1: Illustration of the anchor filtering process when mapping with partial alignments.

Typically, the *evaluation* and *filtering* steps are achieved through the application of already existing approaches from the field of ontology mapping. The *filtering* step can be performed by simply applying a threshold to the score set S, with the threshold value set by a domain expert or learned using a training set. To evaluate the anchors, one can utilize any available concept similarity metric (Shvaiko and Euzenat, 2005). However, such metrics are unfortunately susceptible to concept heterogeneities, where a concept pair for which a human would immediately conclude that it denotes the same information would result in a low similarity values. Such heterogeneities can be mitigated through the combination of multiple similarity metrics, though the aggregation of several similarity values has its disadvantages. For example, given two concept pairs which respectively receive the similarity values

$\{0, 1\}$ and $\{0.5, 0.5\}$ as determined by two metrics, one would be more inclined to accept the first pair than the second, since it can occur that the feature on which a similarity metric relies might be absent while at the same time the maximum score of a given metric is only rarely a false positive. Computing the aggregate of two similarities would thus obscure this information. The approach presented in this paper attempts to tackle this problem by proposing a new way in which a similarity metric can be used to evaluate anchors.

6.2 Proposed Approach

A similarity metric can produce a small set of reliable correspondences, given a sufficiently high similarity threshold. Established matching systems, such as LogMap (Jiménez-Ruiz and Cuenca Grau, 2011) or Anchor-FLOOD (Seddiqui and Aono, 2009), utilize this property to generate a series of anchors on-the-fly to serve as the basis of their general mapping strategy. However, it can be the case that the provided partial alignment originates from a source that is unknown to the mapping system, e.g. a domain expert or a different mapping system in the case where multiple systems are composed in a sequential order. In this case, one can generate a set of reliable correspondences in a way similar to LogMap or Anchor-FLOOD, and utilize this set to evaluate the provided anchors. For example, LogMap (Jiménez-Ruiz *et al.*, 2012b) generates this set by applying a terminological similarity with a strict cut-off. Given a partial alignment from an unknown source, generating a separate set of reliable correspondences presents us with the opportunity to evaluate the partial alignment using this generated set.

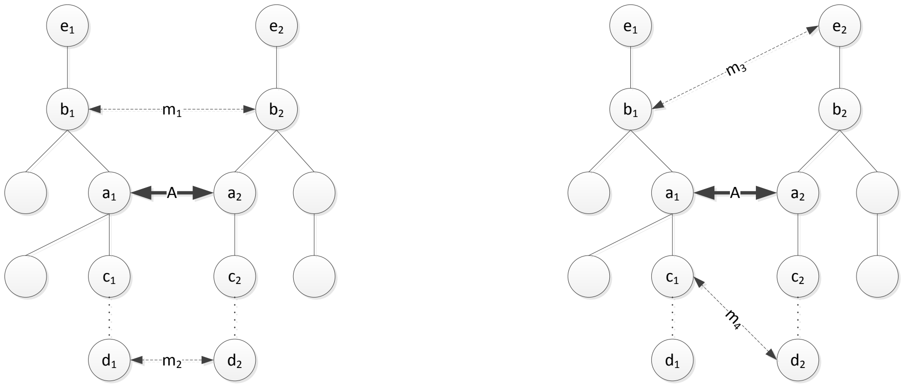
Given an anchor $a_x \in \{a_1, a_2, \dots a_n\}$ and set of generated correspondences $C = \{c_1, c_2, \dots, c_m\}$, in order to evaluate a with C we need a metric for comparing a_x with every element of C. An aggregation of the results for each element of C could then determine whether a_x is filtered from the partial alignment. To compare a_x with any correspondence c_y, it is preferable to have a metric that produces consistent results independent of the taxonomical distances between a_x and c_y within the given ontologies O_1 and O_2. This is to ensure the robustness of the approach in the case that none of the correspondences of C are closely related to a_x.

As an example, let us assume we are comparing an anchor a_1, denoting the concept *car* with two correct correspondences c_1 and c_2, with c_1 denoting the concept *vehicle* and c_2 denoting the concept *physical object.* Both c_1 and c_2 are correct correspondences, hence it would be preferable if the two comparisons with a_x would result in the same value despite *car* being more closely related to *vehicle* than to *physical object.*

One can interpret such a measure as expressing how well an anchor aligns with a correspondences, as opposed to measuring the semantic similarity between the anchor concepts. A correct anchor would thus be expected to be better aligned with respect to a reliably classified correspondence as opposed to an incorrect anchor. To minimize the effect of outliers and utilize all available reliably classified correspondences, one should measure the degree of alignment of an anchor and all given correspondences, and measure how well this measure correlates with the expected result. A way to measure how well an anchor aligns with a given correspondence would be to compute the concept similarities between the concepts in the anchor and the concepts of the given correspondence and express how these similarities differ. To measure this difference in similarity between the concepts of an anchor and the concepts of a given correspondence, we propose a measure of dissonance. Given a correspondence $\{c_1, c_2\}$, an anchor $\{a_1, a_2\}$ and a base similarity measure $sim(a, b) \in [0, 1]$, we define the dissonance d as follows:

$$d(\{c_1, c_2\}, \{a_1, a_2\}) = |sim(c_1, a_2) - sim(c_2, a_1)| \tag{6.1}$$

Using the measure of dissonance, the core of the approach consists of comparing the given anchor to a set of reliably generated correspondences, correct and incorrect, and quantifying to what extend the anchor aligns with the given correspondences. Based on this quantification, the set of anchors can then be filtered. For this research, we will investigate three different metrics when used as base similarity sim.

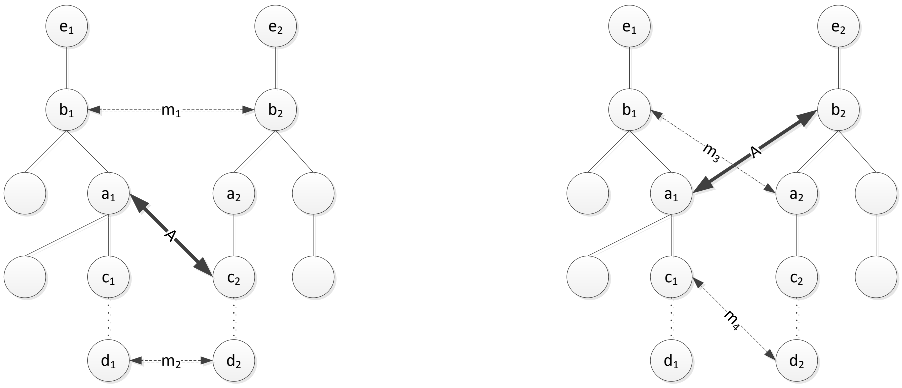


(a) Correct anchor A contrasted against two correct matches m_1 and m_2.

(b) Correct anchor A contrasted against two incorrect matches m_3 and m_4.

Figure 6.2: Example scenarios of an anchor A being compared to either correct matches, illustrating the expected semantic difference between anchors and given correspondences.

To illustrate the principle behind the approach, consider the examples illustrated in Figures 6.2 and 6.3. Each example illustrates two ontologies, an anchor A and two correspondences linking two other concept pairs. Figure 6.2a depicts a correct anchor and two correct correspondences $m_1 = [b_1, b_2]$ and $m_2 = [d_1, d_2]$. m_1 is semantically more related to A than to m_2, thus it can be expected that when calculating $sim(a_1, b_2)$ and $sim(a_2, b_1)$ results in higher values than when computing $sim(a_1, d_2)$ and $sim(a_2, d_1)$. It is reasonable to presume that $sim(a_1, b_2)$ and $sim(a_2, b_1)$ will result in equally high, and $sim(a_1, d_2)$ and $sim(a_2, d_1)$ will result in equally low values, meaning that computing the dissonance $d(m_1, A)$ and $d(m_2, A)$ will result in equally low values, indicating a high degree of alignment.



(a) An incorrect anchor A contrasted against two correct matches m_1 and m_2.

(b) An incorrect anchor A contrasted against two incorrect matches m_3 and m_4.

Figure 6.3: Four example scenarios of an anchor A being compared to incorrect matches, illustrating the irregularity in the expected semantic difference between anchors and given correspondences.

Comparing a correct anchor to dissimilar correspondences is expected to not exhibit this behaviour. Figure 6.2b illustrates a correct anchor A, consisting of the concepts a_1 and a_2, and two incorrect matches m_3 and m_4, which link the concepts b_1 with e_2 and c_1 with d_2 respectively. In this situation, a similarity calculation between a_2 and b_1 is likely to result in a higher value than the similarity between a_1 and e_2. Similarly, the concept similarity between the concepts of A and m_3 are also likely to differ, despite m_4 being semantically more apart from A than m_3.

When given an incorrect anchor, the similarity difference between the concepts of A and the concepts of either correct or incorrect matches are less likely to be predictable, as illustrated in Figure 6.3a and 6.3b. Figure 6.3a depicts an incorrect anchor A being compared to two correct correspondences. Here, both correspondences contain one concept, b_1 and d_2 respectively, which are semantically closer to A than their other concept. Thus, computing a similarity measure between the concepts of a correct correspondence and the concepts of an incorrect anchor will likely produce unequal results, regardless of the semantic distance of the correspondence to the anchor. However, to which degree these similarity will differ is not predictable, since this depends on how semantically related the concepts of the incorrect anchor are. If one were to compare an incorrect anchor to an incorrect correspondences, then the expected difference in concept similarities is not predictable at all, as illustrated in Figure 6.3b. The comparison of A with m_3 is likely to produce a low difference in similarity, being the result of the comparison of a_1 with a_2 and b_1 with b_2. On the other hand, it is also possible the similarity difference can be very large, as illustrated with m_4.

6.2.1 Filtering using Feature Selection

Having identified a measurement which leads to predictable behaviour for correct anchors and less predictable behaviour for incorrect anchors, one now needs to find a method for quantifying this predictability. As previously stated, in order for the dissonance to behave in a predictable way one must use correspondences of which their truth value is known with a high degree of certainty. The correct and incorrect comparison correspondences need to be generated reliably, such that labelling them as true and false respectively results in only few incorrect labels. Assuming that these generated correspondences have indeed their corresponding labels, one can interpret the different dissonance measures as separate samples over a feature space. Given a set of n input anchors $A = \{a_1, a_2, \dots, a_n\}$ and the set of generated correspondences $C = \{c_1, c_2, \dots, c_m\}$ with their respective labels $Y = \{y_1, y_2, \dots, y_m\}$, containing both reliably correct and incorrect correspondences, each correspondence c_x would thus consist of n dissonance measurements $d_{x,i} (i = 1, \dots n)$ and its label y_x. If an anchor a_x is correct, then evaluating the dissonances over C would lead to discernible differences for correct and incorrect correspondences, making the variable representing a_x in the feature space a good predictor of the labels Y.

To determine how well each dimension can serve as a predictor, one can utilize established feature selection techniques (Guyon and Elisseeff, 2003), which have become part of a set of important pre-processing techniques facilitating the use of machine learning and data-mining techniques on high-dimensional datasets. These techniques quantify how much a feature can contribute to the classification of a given labelled dataset. Their scores are then used in order to dismiss features which do not hold information that is relevant for classifying the data, allowing for the reduction of the feature space and the quicker training and execution of classifiers.

For this research, we will use the computed feature scores as evaluation metric for their corresponding anchors. Based on these values, a filtering policy can then dismiss anchors which are unlikely to be correct. Feature selection methods can utilize different underlying principles, for instance using correlation measures or information theory approaches. In order to not bias our approach to a single method, we will evaluate six different feature evaluation measures.

Pearson Correlation Coefficient A fundamental method in the field of mathematical analysis, the *Pearson Correlation Coefficient* (Myers, Well, and Lorch Jr., 2010) measures the linear correlation between two variables. Having the sample set X and Y of two variables, the *Pearson Correlation Coefficient* is defined as:

$$r = \frac{\sum_{i=1}^{n}(X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum_{i=1}^{n}(X_i - \bar{X})^2}\sqrt{\sum_{i=1}^{n}(Y_i - \bar{Y})^2}} \tag{6.2}$$

Spearman Rank Correlation The *Spearman Rank Correlation* (Myers *et al.*, 2010) is a method which utilizes the method of computing the *Pearson Correlation Coefficient.* Here, the sample sets X and Y are transformed into the ranking sets x and y. The correlation between x and y is then computed as:

$$p = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}} \tag{6.3}$$

Gain Ratio Information theoretical approaches have also been employed as measures of feature quality. Information gain techniques compute how much impurity is left in each split after a given attribute has been employed as the root node of a classification tree (Quinlan, 1986). To measure this impurity, the measure of entropy is commonly employed. The entropy of a variable X is defined as:

$$H(X) = -\sum_{x_i} p(x_i)log_2 p(x_i) \tag{6.4}$$

The entropy after observing another variable is defined as:

$$H(X|Y) = -\sum_{y_j} p(y_j) \sum_{x_i} p(x_i|y_j)log_2 p(x_i|y_j) \tag{6.5}$$

The information gain of X is defined as the additional amount of information left after partitioning for all values of Y:

$$IG(X|Y) = H(X) - H(X|Y) \tag{6.6}$$

The *Gain Ratio* is defined as the normalized information gain:

$$GainRatio(X|Y) = IG(X|Y)/H(X) \tag{6.7}$$

Symmetrical Uncertainty The *Symmetrical Uncertainty* (Flannery *et al.*, 1992) is a measure that is similar to the *Gain Ratio*. It however employs a different normalization principle to counteract the bias towards larger attribute sets. Using equations 6.4 and 6.6, the *Symmetrical Uncertainty* $SU(X)$ can be computed as follows:

$$SU(X) = 2\left[\frac{IG(X|Y)}{H(X) + H(Y)}\right] \tag{6.8}$$

Thornton's Separability Index Instead of using a correlation measure, *Thornton's Separability Index*(Thornton, 1998) expresses separability between the classes in a dataset. Specifically, it is defined as the fraction of data-points whose nearest neighbour shares the same classification label. It is computed as follows:

$$TSI = \frac{\sum_{i=1}^{n}(f(x_i) + f(x_i') + 1) \mod 2}{n} \tag{6.9}$$

where f is a binary value function returning 0 or 1, depending on which class label is associated with value x_i. x_i' is defined as the nearest neighbour of x_i.

Fisher's Linear Discriminant *Fisher's Linear Discriminant* (Fisher, 1936) evaluates the discriminatory quality of a set of features by calculating the difference of means of the features and normalizing this distance by a measure of the within-class scatter. The dataset is transformed into a linear space using the projection w which optimizes the output of the value function. The discriminant of two features can be computed as follows:

$$J(w) = \frac{|\mu_{y_1} - \mu_{y_2}|}{s_{y_1}^2 + s_{y_2}^2} \tag{6.10}$$

where μ_y and s_y^2 denote the means and variance of class y.

Using these feature evaluation methods one can evaluate the given anchors of a partial alignments with respect to their discriminatory qualities over the dissonance feature space. Based on the evaluation values, a filtering policy can then decide which anchors to discard before continuing the mapping process. The computation of these measures has been facilitated using the Java-ML framework (Abeel, Van de Peer, and Saeys, 2009).

6.3 Evaluation

To evaluate the proposed technique of filtering anchors, we utilized the *conference* dataset originating from the 2013 Ontology Alignment Evaluation Initiative (Grau *et al.*, 2013). This dataset contains matching tasks, including reference alignments, of real-world ontologies describing the domain of scientific conferences. While this dataset does not contain predefined partial alignments as additional input, it is possible to simply generate partial alignments from the supplied reference alignments. For this domain, it is preferable that the partial alignment also contains incorrect anchors such that the capability of filtering these incorrect anchors can be adequately tested. For each mapping task, *PA* is generated randomly such that it exhibits a *precision* and *recall* of 0.5 with respect to the reference alignment. Since we assume that a similarity metric can produce limited set reliable correspondences given a high threshold, as mentioned in Section 6.2, we limit the set of correct correspondences in the partial alignment to correspondences which do not exhibit a high pairwise similarity. The experiments thus provide an insight to what extent we can reliably evaluate anchors for situations where a basic similarity-based evaluation produces unreliable results.

Each task is repeated 100 times and the results aggregated in order to minimize random fluctuations. For each task, the given approach evaluates the given anchors, such that from the resulting scores a ordered ranking is created. While in a real-world application a given filtering approach would discard a series anchors based on a given policy, for instance by applying a threshold, for an experimental set-up it is more appropriate to perform a precision vs. recall analysis. Such an analysis allows for a comparison of performances without having to limit oneself to a set filtering policies.

To evaluate the dissonance between an anchor and a comparison correspondence, as stated in Section 6.2, a base similarity metric *sim* is required. We investigate three different categories of base similarity metrics:

Syntactic A comparison between concept names and labels using a specific algorithm. The Jaro (Jaro, 1989) similarity was applied for this purpose. Subsection 6.3.1 will present the results of this experiment.

Structural A comparison between concepts which also includes information of related concepts in its computation. As an example of a structural similarity, a profile similarity (Qu *et al.*, 2006) has been evaluated. A profile similarity gathers syntactical information, e.g. concept names, labels and comments, from a given concept and its related concepts into a collection, which is referred to as profile. The similarity of two profiles determines the similarity of the corresponding concepts. The results of this experiment will be presented in subsection 6.3.2.

Lexical A similarity of this type aims to identify the meanings of concept senses within a lexical resource. The senses of the lexical resource are related with each other using semantic relations, e.g. *'is-a-kind-of'* relations, forming a taxonomy of senses. Concept similarities are determined by identifying the correct concept senses and determining the distance of these senses within the lexical taxonomy. This distance is then transformed into a similarity metric. For this evaluation a lexical similarity using WordNet as a lexical resource has been evaluated (Schadd and Roos, 2014b), as described in Chapter 4. Specifically, lsm_2 has been applied as similarity metric and *A-MEAN* as disambiguation policy. The results of this evaluation will be presented in subsection 6.3.3 .

The final score of each anchor is determined by computing the pairwise similarity of the anchor concepts, also computed using *sim*, and multiplying this similarity with the anchor consistency score as determined using the proposed approach, using one of the tested feature evaluation methods. We will compare the rankings of our approach with a *baseline*, which is obtained by computing the pairwise similarities of the anchor concepts using the base similarity *sim*, while omitting the evaluation of the anchors using our approach. The comparison with the *baseline* allows us to establish how much our approach contributes to the evaluation of the given anchors.

The presented approach requires a method of generating the set of correspondences C which serve as individuals of the feature space. In order to apply feature selection techniques on a dataset, the class labels y of each individual must be known, and ideally also correct. Since a single similarity metric can produce a reliable set or correct correspondences, albeit limited in size, one can use this set as the part of C which represent true correspondences. In order to generate reliably incorrect correspondences, one can simply select two concepts at random while ensure that their pairwise similarity is below a threshold. For the experiments, the quantity of incorrect correspondences is set to be equal to the quantity of reliably correct correspondences. To generate C the *Jaro* similarity with thresholds 0.75 and 0.3 was utilized to ensure that the correspondences had a sufficiently high or low similarity.

6.3.1 Syntactic Similarity

In the first performed experiment the *Jaro* similarity was evaluated when applied as *sim* in order to evaluate a syntactical similarity. The generated anchors are evaluated and ranked according to their evaluation scores. We evaluate these rankings by computing their aggregated interpolated precision vs. recall values, displayed in Figure 6.4.

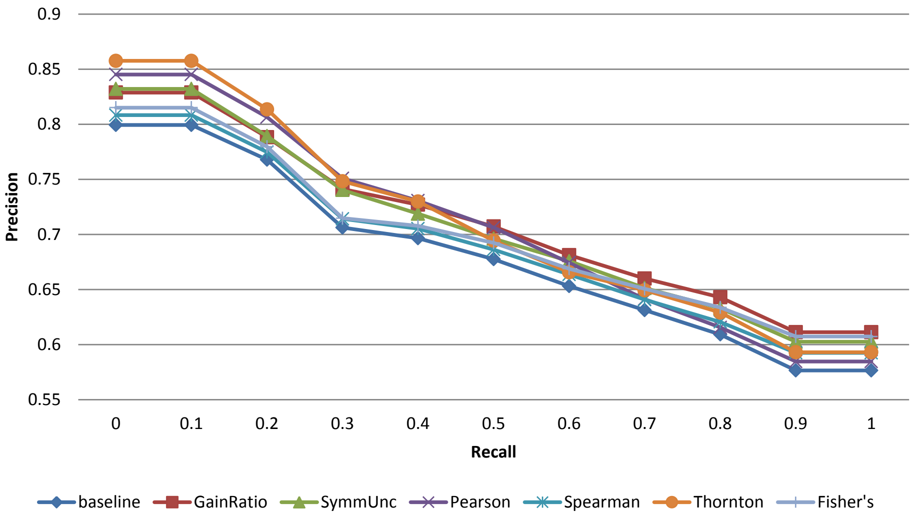


Figure 6.4: Precision vs. recall of the rankings created using a syntactic similarity weighted by the evaluated feature selection methods. The un-weighted variant of the syntactic similarity is used as baseline.

From the results depicted in Figure 6.4 several observations can be made. The most striking observation to be made is that all six tested feature evaluation methods produced a better ranking than the un-weighted baseline. At low recall levels this resulted in an increased precision of up to .057. At the higher recall levels we observe an increase in precision of up to .035.

With regard to the individual feature evaluation metrics a few trends are observable. First of all, we can see that the information theoretical approaches, meaning the *GainRatio* and the *Symmetrical Uncertainty* improve the precision fairly consistently across all recall levels. On average, these measure improve the precision by approximately 0.3. The *Spearman rank correlation* and *Fisher's discriminant* only display a marginal improvement for lower recall levels, however show a more significant improvement for higher recall levels. The most significant improvements for the lower recall levels are observed when applying *Thornton's separability index* and *Pearson's correlation coefficient.*

6.3.2 Structural Similarity

For the second evaluation of our approach, we replaced the *Jaro* similarity with a *profile* similarity for *sim*. The *profile* similarity (Qu *et al.*, 2006) compiles meta-information, primarily the name, comments and annotations, of a given concept and concepts that are linked to the given concept using relations such as *'is-a'* and *'domain-of'*. A profile similarity can be classified as a structural similarity due the utilization of information originating from related concepts. The gathered meta-information is represented as a weighted document-vector, also referred to as a profile. The similarity between two concepts is determined by computing the cosine similarity of their corresponding document vectors. The results of evaluating our approach using a profile similarity as *sim* can be seen in Figure 6.5.

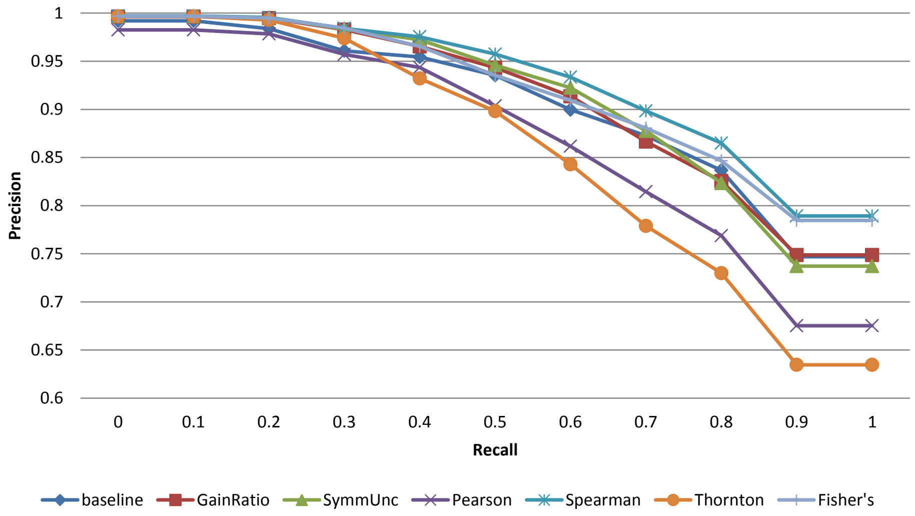


Figure 6.5: Precision vs. recall of the rankings created using a structural similarity weighted by the evaluated feature selection methods. The un-weighted variant of the structural similarity is used as baseline.

From Figure 6.5 we can observe a more mixed result compared to the previous evaluation. The information-theoretical methods, namely *Gain Ratio* and *Symmetrical Uncertainty* outperform the baseline at lower recall levels, maintaining a near-perfect precision of 0.99 for one additional recall level and outperforming the baseline by a margin of roughly .022 at a recall of 0.3. However, for higher recall levels this margin drops until both measures perform roughly on par with the baseline at the highest recall levels. *Thornton's Separability Index* outperforms the baseline only at lower recall levels, while *Pearson's correlation coefficient* performs lower than the baseline. The most promising measures in this experiment were *Fisher's linear discriminant* and the *Spearman rank correlation*, which performed higher than the baseline for all recall levels. Contrary to the baseline, both measures produce a near perfect ranking of 0.99 at a recall of 0.2. The *Spearman rank correlation* produces rankings which have an increased precision of roughly .025 for most recall levels, while for the highest recall levels this difference is widened to roughly .045.

6.3.3 Lexical Similarity

In the third performed evaluation, we evaluated our approach when utilizing a lexical similarity as *sim*. A lexical similarity derives a similarity between two concepts by identifying their intended senses within a corpus and computing the semantic or taxonomic distance between the senses. The resulting distance value is then transformed into a similarity measure. For a lexical similarity to functions it is necessary that the given corpus also models the domains of the two input ontologies. To ensure this, WordNet (Miller, 1995) has been utilized as corpus, which aims at modelling the entire English language. The result of utilizing a lexical similarity as *sim* can be seen in Figure 6.6.

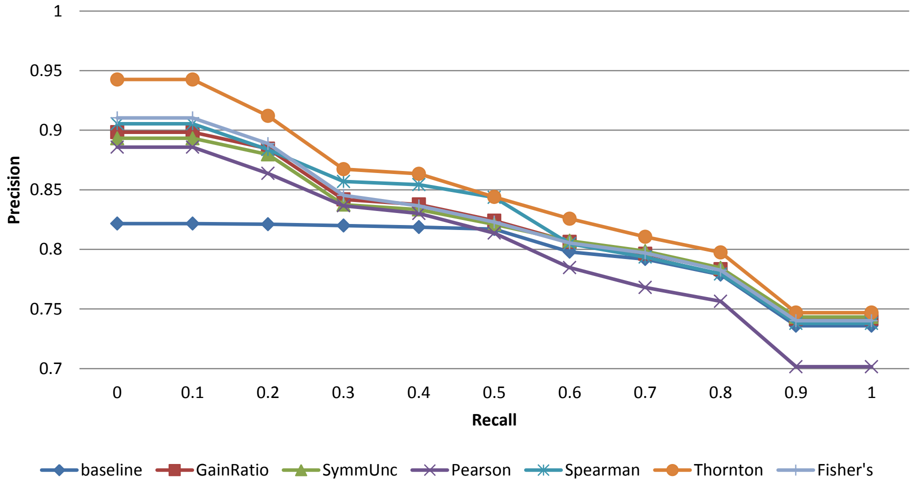


Figure 6.6: Precision vs. recall of the rankings created using a lexical similarity weighted by the evaluated feature selection methods. The un-weighted variant of the lexical similarity is used as baseline.

From Figure 6.6 several key observations can be made. First of all, the baseline displays a distinctively constant precision of .82 up to a recall level of .5. For the lower recall levels, our approach outperforms the baseline by a significant margin using any of the tested feature evaluation methods. Most measures produced an interpolated precision and recall of approximately .9, indicating an improvement of .08. When increasing the recall levels, the performance of these measures slowly approaches the performance of the baseline, while still staying above it. The exception is *Pearson's correlation coefficient*, which performs lower than the baseline at higher recall levels.

The clearly best performing measure is *Thornton's separability index*, which produced a precision higher than both the baseline and the other measures for all recall levels. At recall levels of .3 and higher Thornton's separability index improved upon the baseline by up to .047. At recall levels of .0 and .1 Thornton's separability index produced rankings with a precision of approximately .94, an improvement of .12

compared to the baseline. At a recall level of .2 it still produced rankings with a commendable precision of .91, which is .09 higher than the baseline.

Improvements of this magnitude are particularly important for the utilization of partial alignments, since they allow a significantly larger amount of anchors to be utilized while maintaining a degree of certainty that the anchors are correct. An approach which utilizes partial alignments relies on the quantity and quality of the anchors, but is likely biased towards the quality of the anchors. Thus in order to perform well, such an approach is likely to enforce stringent criteria on the given anchors instead of risking wrong anchors to be included in its computation. In the case of using a lexical similarity to achieve this, our approach would lead to a significantly higher amount of correct anchors being retained.

6.4 Chapter Conclusion and Future Research

We end this chapter by summarizing the results of the experiments (Subsection 6.4.1) and giving an outlook on future research (Subsection 5.4.2) based on the findings presented.

6.4.1 Chapter Conclusions

In this chapter we tackled *research question 3* by proposing a feature-selection-based approach for the evaluation of the given anchors. Our approach is designed to create a feature space where every feature is representative to a specific anchor. We populate this feature space by generating a set of reliably classified correspondences and computing a measure of dissonance for every feature, which measures how well an instance aligns with a given anchor. The intuition behind this approach is that correct anchors should align better with the reliably classified correspondences, which is measured by how well the features of the feature space can serve as predictor of a potential classification task. We apply established feature selection techniques from the field of machine learning to measure the predictor capability of every feature.

We evaluated our approach on the OAEI conference dataset using three different similarities for the computation of the dissonance measures and six different feature evaluation techniques. In the first experiment, we evaluated our technique when applying a syntactic similarity (6.3.1). We concluded from this experiment that all tested feature selection techniques produces consistently better rankings than the evaluated baseline ranking, with some techniques producing a significant improvement. In the next experiment, we evaluated our technique when applying a structural similarity instead (6.3.2). We observed more varied results, with the application of *Fisher's Linear Discriminant* and *Spearman's Rank Correlation* resulting in an improvement in ranking quality compared to the baseline, while the application of the *Pearson Correlation Coefficient* and *Thornton's Separability Index* resulted in a decrease in quality. We concluded that for the rankings of a similarity metric, which itself to an extent can produce reliable rankings, can still be improved through the application of our approach. In the last experiment, we evaluated the performance when applying a lexical similarity (6.3.3). We observed a significant improvement in ranking quality at lower to medium recall levels for all evaluated feature selection techniques. Particularly, the application of the *Thornton's Separability Index* resulted in a ranking quality that is distinctively higher than the other tested techniques and significantly higher than the baseline.

Taking the experimental observations into account, we conclude that the proposed evaluation method is capable of improving the performance of every tested base similarity. Further, we observe that the measured improvements were significant when utilizing a syntactic or lexical base similarity. Combining this observation with the performances of the different feature evaluation techniques leads us to conclude that, prior to integration into a matching system, an evaluation of the chosen configuration is necessary in order to ensure the desired performance improvement.

6.4.2 Future Research

We propose two directions of future research based on our findings presented in this chapter.

(1) The current evaluation was performed on the conference dataset. This dataset does not suffer from terminological disturbances or omissions, such that similarities which utilize terminological information can be applied. Future research could evaluate the approach on the benchmark dataset to investigate its robustness and possibly propose techniques to improve the overall robustness.

(2) The presented technique is focused on the evaluation of anchors. However, another part of the overall filtering procedure is the application of a filtering step which utilizes the evaluation scores. Future research could propose and evaluate different policies. Furthermore, given one or more filter policies one could establish the difference in alignment quality when applying a partial alignment matcher with or without a complete anchor filtering procedure (i.e. executing the proposed evaluation approach and applying a filter policy).

Chapter 7

Anchor-Based Profile Enrichment

This chapter is an updated version of the following publication:

1. Schadd, Frederik C. and Roos, Nico (2015). Matching Terminological Heterogeneous Ontologies by Exploiting Partial Alignments. *9th International Conference on Advances in Semantic Processing (SEMAPRO 2015)*, pp 13−18, *Best Paper Award Winner.*

The previous chapters have dealt with profile similarities on multiple occasions. In Chapter 4 we introduced a profile-based method for concept-sense disambiguation and in Chapter 5 we introduced a profile-based method for matching ontologies with partial alignments. Both chapters attempt to improve the exploitation of external resources, being lexical resources and partial alignments respectively, using profile-based methods. In this chapter we approach the combination of profiles and external resources from a different perspective. Here, we attempt to improve an existing resource-independent metric, specifically a profile similarity, through the addition of a provided resource.

Profile similarities are widely used in the field of ontology mapping. Despite being a type of syntactical similarity, exploiting the terminological information of ontology concepts, they are fairly robust against terminological heterogeneities due to the large scope of information that is exploited for each concept. For instance, if two corresponding concepts A and B have very dissimilar labels, then a profile similarity can still derive an appropriate similarity score if other information close to A and B still matches, e.g. their descriptions or parent's labels. However, this robustness has a limit. For ontology pairs between which there is a significant terminological gap, it is less likely that there is matching information in the vicinities of two corresponding concepts A and B. Hence, a typical profile similarity is unsuited to deal with this kind of matching problem. A sophisticated mapping system would then configure itself such that the profile similarity would not be used at all.

This chapter addresses the fourth research question by exploiting a provided partial alignment such that a profile similarity can be used to match ontologies between which there exists a significant terminological gap. Given two ontologies O_1 and O_2, we redefine the neighbourhood of a concept, from which information for a profile is gathered, by including the semantic relations of the partial alignment to the set of exploitable relations. For example, given a concept $a_1 \in O_1$, a concept $b_1 \in O_1$ which is in the neighbourhood of a_1 and a partial alignment PA containing a correspondence c which matches b_1 to a concept in O_2, it is possible to exploit c in order to add information originating from O_2 to the profile of a_1, thus using the terminology of the other ontology and bridging the terminological gap between O_1 and O_2. We evaluate our approach on a subset of the OAEI benchmark dataset and the OAEI multifarm dataset, representing two datasets consisting solely of matching problems with significant terminological gaps.

The remainder of this chapter is structured as follows. Section 7.1 discusses work related to profile similarities and partial alignments. Section 7.2 introduces profile similarities in more detail and illustrates the terminological gap with an example. Section 7.3 details our approach and Section 7.4 presents the results and discussions of the performed experiments. Finally, Section 7.5 presents the conclusions of this chapter and discusses future work.

7.1 Related Work

Profile similarities have seen a rise in use since their inception. Initially developed for the Falcon-AO system (Qu *et al.*, 2006), this type of similarity has been used in ontology mapping systems such as AML (Cruz *et al.*, 2009), RiMoM (Li *et al.*, 2009) and CroMatcher (Gulić and Vrdoljak, 2013). These systems typically apply the same scope when gathering information for a concept profile, being the parent concepts and children concepts. Some systems, such as YAM++ (Ngo *et al.*, 2012), limit the scope to the information contained in the concept annotations and labels.

There exist some works which aim at extending the scope of exploited profile information in order to improve the effectiveness of the similarity. The deployed profile similarity in the mapping system PRIOR (Mao *et al.*, 2007) extends the scope of exploited information to the grand-parent concepts and grand-children concepts, thus providing a larger amount of exploitable context information.

7.1.1 Semantic Enrichment

One way in which additional information can be exploited is through *semantic enrichment*. *Semantic enrichment* describes any process which takes any ontology O as input and produces as output the enhanced ontology $E(O)$, such that $E(O)$ expresses more semantic information than O. For this purpose, a semantic enrichment process typically exploits resources such as stop-word lists, allowing to identify words as stop-words, or lexical resources, allowing to annotate words or concepts with their lexical definitions. The disambiguation procedure of subsection 4.3 can serve as an example of an enrichment process which enriches concepts with lexical senses. However, unlike in subsection 4.3, a formal enrichment process typically separates the enrichment process with the subsequent similarity computations.

Semantic enrichment has been applied to ontology mapping in a non-profile context. Notable examples are the addition of synonyms to the concept descriptions by exploiting lexical resources. An example of this is the LogMap system (Jiménez-Ruiz and Cuenca Grau, 2011), which is capable of adding information from WordNet or UMLS to the ontologies prior to mapping. Another example is YAM++ (Ngo *et al.*, 2012) which uses a machine translator to generate English translations of labels prior to mapping.

A noteworthy application of semantic enrichment for a profile similarity is the work by Su and Gulla (2004). Here, the semantic enrichment process exploits a corpus of natural language documents. Using a linguistic classifier and optional user input, the corpus documents are assigned to the ontology concepts, such that each assignment asserts that the ontology concept is discussed in its associated document. The exploitation of the corpus documents results in the concept profiles containing much more terms, which is particularly beneficial for matching tasks where the ontologies contain only little terminological meta-information. The concept profiles are constructed as feature-vectors using the *Rocchio algorithm* (Aas and Eikvil, 1999), where a feature-vector describing the concept c is specified as the average feature-vector over all documents d_i that contain the concept c. The similarities between concepts are determined by computing the cosine-similarity between their feature-vectors.

7.2 Profile Similarities and the Terminological Gap

Profile similarities are a robust and effective type of similarity metric and deployed in a range of state-of-the-art ontology matching systems (Qu *et al.*, 2006; Cruz *et al.*, 2009; Li *et al.*, 2009). They rely on techniques pioneered in the field of information retrieval (Manning *et al.*, 2008), where the core problem is the retrieval of relevant documents when given an example document or query. Thus, the stored documents need to be compared to the example document or the query in order to determine which stored documents are the most similar and therefore returned to the user. A profile similarity adapts these document comparison techniques by constructing a virtual document for each ontology concept, also referred to as the *profile* of that concept, and determines the similarity between two concepts x and y by comparing their respective profiles. The core intuition of this approach is that x and y can be considered similar if their corresponding profiles can also be considered similar.

As their origin already implies, profile similarities are language-based techniques (Euzenat and Shvaiko, 2007). Language-based techniques interpret their input as an occurrence of some natural language and use appropriate techniques to determine their overlap based on this interpretation. A language-based technique might for instance perform an analysis on the labels of the concept in order to determine their overlap. For instance given the two concepts *Plane* and *Airplane* a language-based analysis of their labels would result in a high score since the label *Plane* is completely contained within the label *Airplane*. Thus, despite the labels being different, a high similarity score would still be achieved. However, the degree of surmountable label-difference has a limit for language-based techniques. The labels of the concepts *Car* and *Automobile* have very little in common with respect to shared characters, tokens or length. Thus, many language-based techniques are unlikely to result in a high value.

Profile similarities have the advantage that they draw from a wide range of information per concept. Thus terminological differences between the labels of two concepts can still be overcome by comparing additional information. This additional information typically includes the comments and annotations of the given concept and the information of semantically related concepts (Qu *et al.*, 2006; Mao *et al.*, 2007). The range of exploited information of a typical profile similarity is visualized in Figure 7.1.

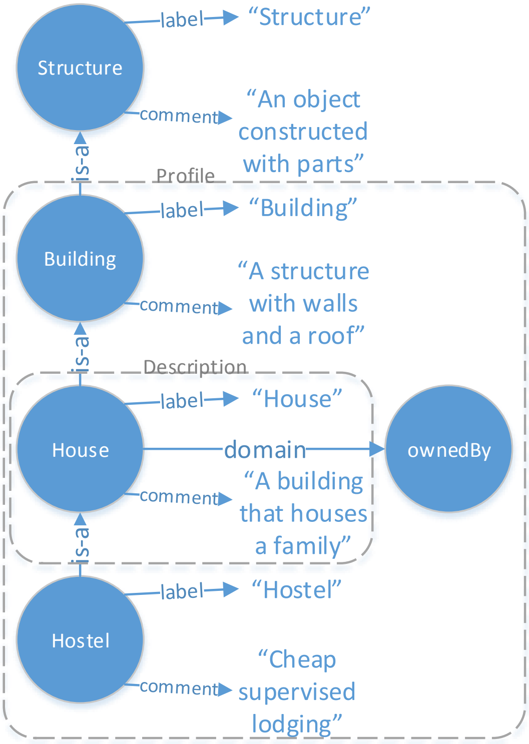


Figure 7.1: Illustration of the typical range of exploited information of a profile similarity.

Figure 7.1 illustrates the range of exploited information when constructing a profile of the concept *House*. The concept itself contains some encoded information in the form of a label and a comment. The unification of this information is also referred to as the *description* of the given concept. The concept *House* is also associated to other concepts through either semantic relations or because it is associated with a property of the other concepts. The information encoded in these related concepts plus the information in the description of the concept *House* make up the information typically found in the profile of *House*.

In order for two profiles to be similar, they must contain some shared terminology. For example, the concepts *House* and *Home* can still be matched if their parents contain the word *Building* or if a concept related *Home* contains the word *"House"*. In essence, in order for profile similarities to be effective it is still required that the two given ontologies O_1 and O_2 exhibit some overlap with respect to their terminologies. However, this is not always the case as two ontologies can model the same domain using a completely different terminology. This can be the result of one ontology using synonyms, different naming conventions or the usage of acronyms. Furthermore, two ontologies might even be modelled in a different language. For example, one might need to match two biomedical ontologies where one is modelled in English and one in Latin. In the multilingual domain, the terminological difference depends on the relatedness between the given languages. For example, ontologies defined using English and French might have some overlap since English has adapted words from the French language and vice-versa. However, such a limited overlap is unlikely to occur when for instance comparing French and Romanian ontologies. Matching tasks with little terminological overlap can still be regarded as difficult, since the overlapping terms might be concentrated in a few sub-structures of the ontologies, meaning that the other sub-structures would exhibit no terminological overlap at all. The terminological gap between two ontologies is illustrated in Figure 7.2.

Figure 7.2 displays the example of Figure 7.1 next to a series of concepts from a different ontology, *Ontology 2*, modelling the same entities. The terminological gap is illustrated through the fact that all information in *Ontology 2* is modelled in German instead of English. As we can see, comparing the concept *House* with its equivalent concept *Haus* using a typical profile similarity is unlikely to produce a satisfying result, since neither the concepts *Haus* and *House* nor their related concepts contain any overlapping terminology. Therefore, additional measures are necessary in order to ensure the effectiveness of profile similarities when the given ontologies have little to no shared terminology.

7.3 Anchor-Based Profile Enrichment

A typical profile similarity is inadequate for ontology matching problems with significant terminological gaps. One way of tackling this issue is through semantic enrichment by exploiting lexical resources such as WordNet (Miller, 1995), UMLS (Bodenreider, 2004) or BabelNet (Navigli and Ponzetto, 2010). Techniques which fall under this category work by looking up each concept in the given resource and adding synonyms, additional descriptions or translations to the concept definition. However, these techniques rely on several assumptions: (1) the availability of an appropriate resource for the given matching problem, (2) the ability to locate appropriate lexical entries given the naming formats of the ontologies, and (3) the ability to disambiguate concept meanings such that no incorrect labels or comments

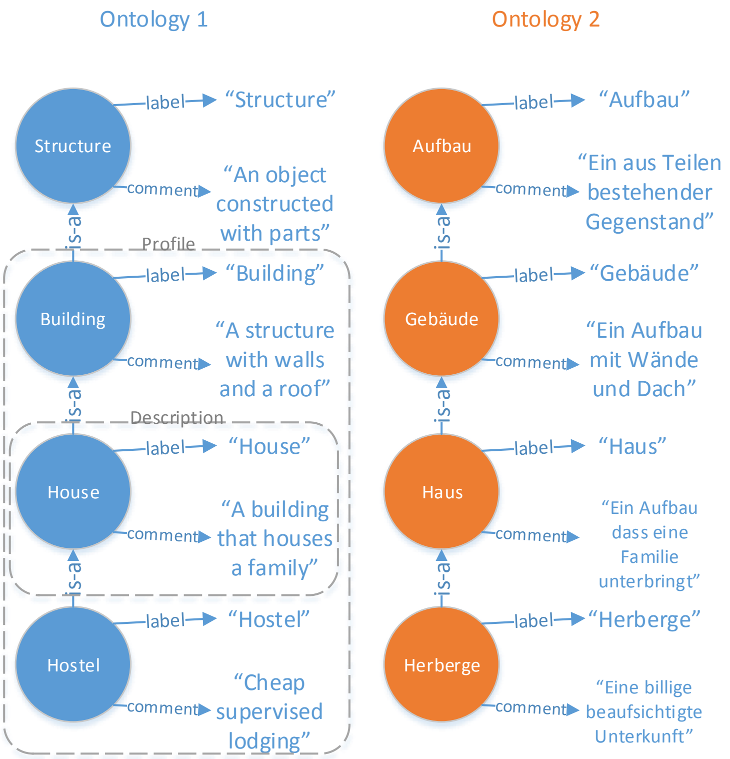

Ontology 1
Ontology 2
Structure
label
“Structure”
comment
“An object constructed with parts”
is-a
Profile
Building
label
“Building”
comment
“A structure with walls and a roof”
is-a
Description
House
label
“House”
comment
“A building that houses a family”
is-a
Hostel
label
“Hostel”
comment
“Cheap supervised lodging”
Aufbau
label
“Aufbau”
comment
“Ein aus Teilen bestehender Gegenstand”
is-a
Gebäude
label
“Gebäude”
comment
“Ein Aufbau mit Wände und Dach”
is-a
Haus
label
“Haus”
comment
“Ein Aufbau dass eine Familie unterbringt”
is-a
Herberge
label
“Herberge”
comment
“Eine billige beaufsichtigte Unterkunft”

Figure 7.2: Illustration of a terminological gap between two ontologies modelling identical concepts.

are added to the concept definition. We can see that the performance of such techniques is severely impacted if any of these assumptions fail. If (1) and (2) fail then it is not possible to add additional information to the concept definition, thus causing the ontology concepts to be compared using only their standard profiles. Examples of assumption (1) failing would be the lack of access to an appropriate resource, for instance due to lack of connectivity, or the lack of existence of any appropriate resource for the given matching task due to the specific nature of the ontologies. As an example for assumption (2) failing, let us consider an example concept *EE* and it's corresponding lexical entry *Energy Efficient*, referring to a type of engine. It is easy to see that the labels are very different, making it a difficult task to match concepts to lexical entries. To ensure the ability of identifying correct lexical entries when dealing with ambiguous concepts, one needs to apply a disambiguation technique, as introduced in sub-section 4.1.2. The current state-of-the-art disambiguation systems can achieve an accuracy of roughly 86% (Navigli, 2009), meaning that even if a state-of-the-art system is applied there is still a significant proportion of concepts which would be associated with unrepresentative information based on incorrectly designated lexical entries.

In the case that an appropriate lexical resource is not available, other measures are necessary to overcome the terminological gap. These typically are the exploitation of other ontological features, for example the ontology structure. However, it may be the case that instead of a lexical resource a different kind of resource is available to be exploited. For a given mapping problem it is possible that an incomplete alignment, also refereed to as *partial alignment*, is available as additional input. A partial alignment can stem from efforts such as a domain expert attempting to create an alignment, but being unable to complete it due to given circumstances, or from a high-precision system generating such an alignment. The correspondences of the given partial alignment can then be exploited in order to determine the unidentified correspondences.

Our approach aims at adapting profile similarities to be appropriate for matching problems with significant terminological gaps through the exploitation of partial alignments. It is based on the insight that an ontology will consistently use its own terminology. For instance, if an ontology uses the term *Paper* to refer to scientific articles, it is unlikely to use the equivalent term *Article* in the descriptions of other concepts instead, especially if the ontology is designed using a design principle that enforces this property (Sure, Staab, and Studer, 2002). However, if a partial alignment contains the correspondence *Paper-Article*, then one can use this insight to one's advantage. For instance, given the concept *Accept_Paper* a profile similarity is more likely to match it to its appropriate counterpart *Approve_Article* if the profile of *Accept_Paper* contains the term *'Article'*.

A partial alignment *PA* is a set of correspondences, with each correspondence asserting a semantic relation between two concepts of different ontologies. The types of relations modelled in a partial alignment, e.g. $\sqsupseteq$, $\perp$, $\sqcap$ and $\equiv$, are typically also modelled in an ontology and thus exploited in the construction of a profile. Thus, by semantically annotating the given ontologies O_1 and O_2 with the correspondences of *PA* it becomes possible to exploit these newly asserted relations for the creation of the concept profiles. This enables us to construct the profiles of O_1 using a subset of the terminology of O_2, increasing the probability of a terminological overlap between the profiles of two corresponding concepts. This idea is illustrated in Figure 7.3.

Before we introduce our approach, we need to define a series terms and symbols that will be used in the following sections:

Collection of words: A list of unique words where each word has a corresponding weight in the form of a rational number.

+: Operator denoting the merging of two collections of words.

×: Operator denoting element-wise multiplication of term frequencies with a weight.

depth(x): The taxonomy depth of concept x within its ontology.

D: The maximum taxonomical depth of a given ontology.

Next, it is necessary to provide a definition of a basic profile similarity upon which we can base our approach. For this, we provide a definition similar to the work

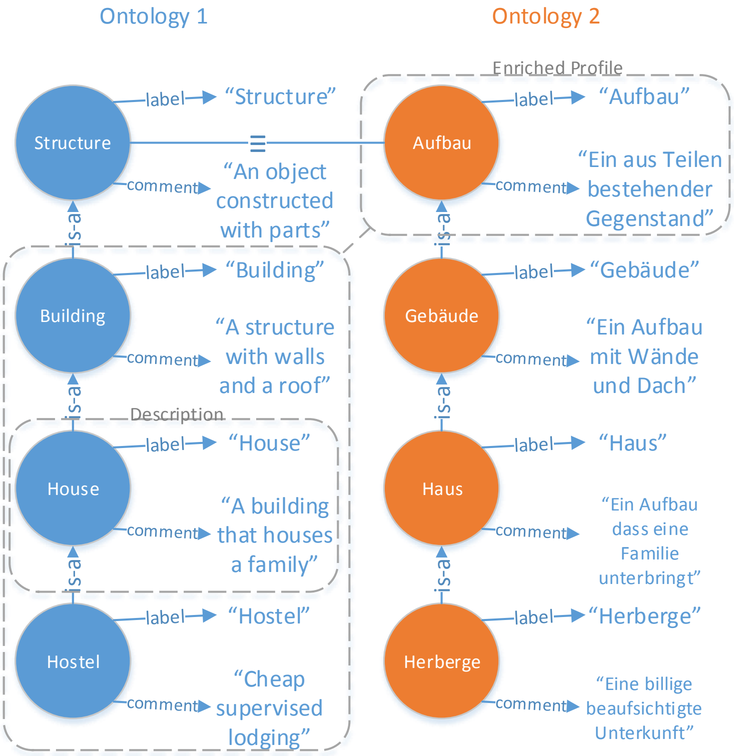

Ontology 1
Ontology 2
Enriched Profile
Structure
label
"Structure"
≡
comment
"An object constructed with parts"
Aufbau
label
"Aufbau"
comment
"Ein aus Teilen bestehender Gegenstand"
is-a
Building
label
"Building"
comment
"A structure with walls and a roof"
Gebäude
label
"Gebäude"
comment
"Ein Aufbau mit Wände und Dach"
Description
House
label
"House"
comment
"A building that houses a family"
Haus
label
"Haus"
comment
"Ein Aufbau dass eine Familie unterbringt"
Hostel
label
"Hostel"
comment
"Cheap supervised lodging"
Herberge
label
"Herberge"
comment
"Eine billige beaufsichtigte Unterkunft"

Figure 7.3: Two equivalent concepts being compared to a series of anchors.

by Mao *et al.* (2007). Neighbouring concepts are explored using a set of semantic relations, such as *isChildOf* or *isParentOf*.

A base function of a profile similarity is the description of a concept, which gathers the literal information encoded for that concept. Let x be a concept of an ontology, the description $Des(x)$ of x is a collection of words defined as follows:

$$\begin{aligned} Des(x) \quad = \quad & \text{collection of words in the name of } x \\ & + \text{collection of words in the labels of } x \\ & + \text{collection of words in the comments of } x \\ & + \text{collection of words in the annotations of } x \end{aligned} \tag{7.1}$$

We define the profile of x as the merger of the description of x and the descriptions of concepts that are semantically related to x :

$$Profile(x) = Des(x) + \sum_{p \in P(x)} Des(p) + \sum_{c \in C(x)} Des(c) + \sum_{r \in R(x)} Des(r) \tag{7.2}$$

where

$$\begin{aligned}P(x) &= \{p|x \ \mathit{isChildOf}\, p\}\\ C(x) &= \{c|c \ \mathit{isChildOf}\, x\}\\ R(x) &= \{r|r \ \mathit{isRelatedTo}\ x \wedge r \notin P(x) \cup C(x)\}\end{aligned}$$

In order to compute the similarity between two profiles, they are parsed into a vector-space model and compared using the cosine similarity (Pang-Ning *et al.*, 2005). Note that while it is possible to weigh the descriptions of the related concepts and the different collections within the description of each concept, similar to the model presented in sub-section 4.3.3. However, we opt for a uniform weighting for two reasons: (1) A detailed analysis of the influence of these weights would provide little research value since this analysis has already been performed in sub-section 4.4.3 for the model of sub-section 4.3.3, and (2) the profile similarity and its variations can be easier understood and replicated when using a uniform weighting scheme.

To bridge the terminological gap we aim to exploit the semantic relations provided by a given partial alignment PA, such that we can enhance the profile of a concept $x \in O_1$ using the terminology of O_2. We refer to this enlarged profile as the *anchor-enriched-profile*. For this, we explore the parents, children and properties of a concept x (or ranges and domains in case x itself is a property). If during this exploration a concept y is encountered which is mapped in a correspondence in PA to a concept $e \in O_2$, then $\mathit{Profile}(x)$ is extended with $Des(e)$.

We will define the set which describes the merged collection of *parentally-anchored-descriptions* (PAD) with respect to concept x in three variations. These gather the descriptions of anchored concepts from the ancestors of x. To measure the improvement caused by the addition of these sets, we also define the omission of any such description. PAD's are defined as follows:

$$\begin{aligned}\mathrm{PAD}_0(x, PA) = \ & \emptyset\\ \mathrm{PAD}_1(x, PA) = \ & \textstyle\sum_{e \in E} Des(e);\ \ \mathit{where}\\ & E = \{e|\exists < id, y, e, t, c > \in PA; y \ \mathit{isAncestorOf}\ x\}\\ \mathrm{PAD}_2(x, PA) = \ & \textstyle\sum_{e \in E} \omega \times Des(e);\ \ \mathit{where}\\ & E = \{e|\exists < id, y, e, t, c > \in PA; y \ \mathit{isAncestorOf}\ x\}\\ & \wedge\ \omega = \tfrac{D - |depth(x) - depth(y)|}{D}\end{aligned} \tag{7.3}$$

An interesting point to note is that PAD_2 utilizes the same set of concepts as PAD_1, but weights each description using its relative distance to x, such that the descriptions of closer concepts receive a higher weight.

Exploring the children of x, we define the merged collection of *child-anchored-descriptions* (CAD) in a similar way:

$$\begin{aligned}\mathrm{CAD}_0(x, PA) = \ & \emptyset\\ \mathrm{CAD}_1(x, PA) = \ & \textstyle\sum_{e \in E} Des(e);\ \ \mathit{where}\\ & E = \{e|\exists < id, y, e, t, c > \in PA; y \ \mathit{isDescendantOf}\ x\}\\ \mathrm{CAD}_2(x, PA) = \ & \textstyle\sum_{e \in E} \omega \times Des(e);\ \ \mathit{where}\\ & E = \{e|\exists < id, y, e, t, c > \in PA; y \ \mathit{isDescendantOf}\ x\}\\ & \wedge\ \omega = \tfrac{D - |depth(x) - depth(y)|}{D}\end{aligned} \tag{7.4}$$

Lastly, we can explore the relations defined by the properties of the ontology, being *isDomainOf* and *isRangeOf*. Defining O_c as the set of concepts defined in ontology O and O_p as the set of properties of O, we define the merged collection of *relation-anchored-descriptions* (RAD) in two variations as follows:

$$
\begin{aligned}
\mathrm{RAD}_0(x,\mathit{PA}) = \;& \emptyset \\
\mathrm{RAD}_1(x,\mathit{PA}) = \;& \begin{cases} \sum_{e \in E} \mathit{Des}(e);\ \mathit{where} & \\ \quad E = \{e \mid \exists < id, y, e, t, c > \in \mathit{PA}; x\ \mathit{isDomainOf}\ y\} & \mathit{if}\ x \in O_c \\ \sum_{e \in E} \mathit{Des}(e);\ \mathit{where} & \\ \quad E = \{e \mid \exists < id, y, e, t, c > \in \mathit{PA}; y\ \mathit{isDomainOf}\ x \vee & \\ \quad y\ \mathit{isRangeOf}\ x\} & \mathit{if}\ x \in O_p \end{cases} \\
\mathrm{RAD}_2(x,\mathit{PA}) = \;& \begin{cases} \sum_{e \in E \cup F} \mathit{Des}(e);\ \mathit{where} & \\ \quad E = \{e \mid \exists < id, y, e, t, c > \in \mathit{PA}; x\ \mathit{isDomainOf}\ y\} & \\ \quad \mathit{and}\ F = \{f \mid \exists < id, y, f, t, c > \in \mathit{PA}\ \exists z \in O_p; & \\ \quad x\ \mathit{isDomainOf}\ z \wedge y\ \mathit{isRangeOf}\ z\} & \mathit{if}\ x \in O_c \\ \sum_{e \in E} \mathit{Des}(e);\ \mathit{where} & \\ \quad E = \{e \mid \exists < id, e, y, t, c > \in \mathit{PA}; y\ \mathit{isDomainOf}\ x \vee & \\ \quad y\ \mathit{isRangeOf}\ x\} & \mathit{if}\ x \in O_p \end{cases}
\end{aligned}
\tag{7.5}
$$

The noteworthy difference between RAD_1 and RAD_2, given a property z which has x as its domain, is that RAD_2 will include the description of the range of y in addition to the description of y itself. As an example, assume we are given the concepts *Car* and *Driver* being linked by the property *ownedBy*. Constructing the anchor-enriched-profile of *Car* using the set RAD_1 would mean that we only investigate if *ownedBy* is mapped in *PA*. Using RAD_2 means we also investigate *Driver*, which could provide addition useful context.

Given a partial alignment *PA* between ontologies O_1 and O_2, and given a concept x, we define the *anchor-enriched-profile* of x as follows:

$$
\mathit{Profile}^{AE}_{\kappa,\lambda,\mu}(x,\mathit{PA}) = \mathit{Profile}(x) + \mathrm{PAD}_\kappa(x,\mathit{PA}) + \mathrm{CAD}_\lambda(x,\mathit{PA}) + \mathrm{RAD}_\mu(x,\mathit{PA}) \tag{7.6}
$$

7.4 Experiments

In this section we will detail the performed experiments to test the effectiveness of our approach and discuss the obtained results. To adequately test our approach we need (1) a dataset with matching problems demonstrating terminological gaps and (2) a partial alignment for each matching task. For this we will utilize the *benchmark-biblio* (Aguirre *et al.*, 2012) and *multi-farm* (Meilicke *et al.*, 2012) datasets. For the benchmark dataset we selected only mapping tasks where the concept names have been significantly altered in order to represent terminological gaps. The multi-farm dataset contains only matching problems with terminological gaps by design, thus requiring no sub-selection of matching tasks. For each task a partial alignment *PA* is supplied by randomly sampling the reference alignment *R*. Each matching task is repeated 100 times with a newly sampled partial alignment in order to mitigate the variance introduced by randomly sampling PA from R. To evaluate the produced alignments we will use the measures of P^*, R^* and F^*, as introduced in Section 2.3. The experiments in the remainder of this section are structured as follows:

- Subsection 7.4.1 details the experiments performed on the benchmark dataset.
- In subsection 7.4.2 the same evaluation is performed on the multi-farm dataset.
- In subsection 7.4.3 we will evaluate some variations of our approach using different measures of $R(PA, R)$ to investigate how the different variations cope with varying sizes of PA
- We compare our approach to complete systems utilizing lexical resources instead of partial alignments when evaluating the multi-farm dataset in subsection 7.4.4. To establish the potential of our approach, this comparison includes one variation where the ontologies are semantically enriched using a lexical resource.

7.4.1 Benchmark

For our first experiment, we evaluate our approach on the *benchmark-biblio* dataset originating from the 2014 OAEI competition (Dragisic *et al.*, 2014). This dataset presents a synthetic evaluation where an ontology is matched against systematic alterations of itself. For this experiment, we focus our attention only on matching tasks with significant terminological gaps, for which our approach is aimed at. Specifically, we evaluate the following matching tasks: 201-202, 248-254, 257-262 and 265-266. These represent matching tasks in which the concept names and annotations have been significantly altered. For each task we randomly sample a partial alignment PA from the reference alignment R with the condition that $R(PA, R) = 0.5$. From the output alignment we compute the adapted Precision, Recall and F-Measure P^*, R^* and F^* and aggregate the results over the entire dataset. We evaluate our approach with this method using every possible combination of κ, λ and μ, which determine to what extent profiles are enriched using anchored concepts (as specified in Equation 7.6). The results of this evaluation can be see in Table 7.1.

First, an important aspect to note is that the configuration $\kappa = 0, \lambda = 0, \mu = 0$, which can also be denoted as $Profile^{AE}_{0,0,0}$, is the baseline performance. This configuration results in the execution of a standard profile similarity, thus not utilizing any relations provided by the partial alignment. The baseline performance of $Profile^{AE}_{0,0,0}$ is very low, with an F^* value of only 0.119, demonstrating how a profile similarity struggles with this type of matching task. By comparing the performance of $Profile^{AE}_{0,0,0}$ to the performance of any other configuration we can see that our approach improves upon the performance of the baseline profile for every tested configuration. Enriching the profiles using only PAD$_1$ or PAD$_2$ resulted in an F-Measure of 0.251 and 0.230 respectively. Increases of a similar amount in F-Measure can also be seen when only using CAD$_1$, CAD$_2$, RAD$_1$ or RAD$_2$. However, we can observe that applying CAD$_1$ or CAD$_2$ has a more significant impact on the Precision, with an increase of up to 0.2 approximately. Applying RAD$_1$ or RAD$_2$ has

κ	λ	μ	P^*	R^*	F^*
0	0	0	0.520	0.093	0.119
0	0	1	0.420	0.208	0.268
0	0	2	0.402	0.215	0.271
0	1	0	**0.719**	0.199	0.277
0	1	1	0.631	0.307	0.398
0	1	2	0.622	0.314	0.401
0	2	0	0.676	0.187	0.251
0	2	1	0.567	0.298	0.380
0	2	2	0.526	0.294	0.368
1	0	0	0.480	0.197	0.251
1	0	1	0.538	0.310	0.375
1	0	2	0.539	0.318	0.383
1	1	0	0.602	0.296	0.372
1	1	1	0.615	0.404	0.476
1	1	2	0.620	**0.409**	**0.482**
1	2	0	0.544	0.287	0.346
1	2	1	0.568	0.391	0.453
1	2	2	0.564	0.393	0.453
2	0	0	0.453	0.186	0.230
2	0	1	0.481	0.298	0.357
2	0	2	0.464	0.292	0.348
2	1	0	0.589	0.278	0.345
2	1	1	0.590	0.386	0.455
2	1	2	0.589	0.384	0.454
2	2	0	0.561	0.272	0.329
2	2	1	0.558	0.377	0.440
2	2	2	0.531	0.363	0.422

Table 7.1: Aggregated adapted Precision, Recall and F-Measure when evaluating all variations of our approach on a selection of tasks from the Benchmark dataset.

a more significant effect on the resulting Recall. Applying only RAD$_2$ for example resulted in an increase in Recall of approximately 0.12.

While applying PAD, CAD or RAD separately did result into an improvement in performance, the resulting alignment quality however does not yet resemble the quality one typically sees when tackling easier matching tasks. This changes when enriching the profiles by exploiting multiple relation types. Enriching the profiles with two different sets of anchored descriptions typically results in an F-Measure of approximately 0.35-0.37. Two combinations however exceed this typical performance range. Applying $Profile^{AE}_{0,1,1}$ and $Profile^{AE}_{0,1,2}$ resulted in an F-Measure of 0.398 and 0.401 respectively. This increase in F-Measure is the result of a comparatively higher Precision compared to other dual-combinations of description sets.

Finally, we can see that utilizing all description sets, being PAD, CAD and RAD, has resulted in the highest measured performance. The best performance of this evaluation has been the result of applying $Profile^{AE}_{1,1,2}$, with a Precision of 0.62, a Recall of 0.409 and an F-Measure of 0.482. Comparing the performance of $Profile^{AE}_{1,1,2}$ with $Profile^{AE}_{2,2,2}$ indicates that equally weighing the descriptions of concepts that are anchored via ancestors or descendants results in a better performance than giving these descriptions a higher weight if they are more closely related to the given concept. This can also be seen when $Profile^{AE}_{0,1,0}$ with $Profile^{AE}_{0,2,0}$, albeit with a less pronounced difference. For variations that differ only with respect to RAD we can see that there is no clear trend indicating whether RAD_1 or RAD_2 performs better. RAD_1 resulted in a better performance if PAD_2 is also applied. However, for combinations which did not utilize PAD_2 the application of RAD_2 resulted in better performances instead. We will investigate in sub-section 7.4.2 whether these findings are consistent.

Overall, we can conclude that our approach indeed enables profile similarities to better cope with matching problems that are characterized by a significant terminological gap. However, while our approach certainly improves the performance for these matching problems, an F-Measure of 0.482 indicates that the approach alone is not yet sufficient to autonomously tackle these tasks.

7.4.2 MultiFarm

In this section we will present the results of our evaluation on the MultiFarm dataset. This data-set stems from the OAEI 2014 (Dragisic *et al.*, 2014) competition. The terminologies of the ontologies in this dataset vary greatly since it is designed to be a cross-lingual dataset[1]. The set consists of 8 ontologies that are modelled using 9 languages (including English). For each pair of ontologies a set of mapping tasks exists consisting of every possible combination of selecting different languages. As during the previous evaluation, we generate the partial alignments by randomly sampling the reference alignment with the condition that $R(PA, R) = 0.5$ and aggregate the results of 100 evaluations for each task. This evaluation is repeated for every possible combination of κ, λ and μ. The result of this evaluation is presented in Table 7.2.

First, by comparing the performance of the baseline configuration $Profile^{AE}_{0,0,0}$ to any configuration of our approach we can easily see that our approach improves upon the performance of the baseline. Adding the sets PAD or CAD using either variation typically resulted in an F-Measure of 0.39-0.43, an improvement of 0.07 to 0.11 when compared to the baseline. Curiously, enriching the profiles using RAD alone typically resulted in a F^* score of approximately 0.5. This could indicate that for this dataset the concept annotations more often contain terms of related concepts than ancestors or descendants.

Looking at dual-combinations between PAD, CAD and RAD we can see a consistent increase in performance. Of these combinations, $Profile^{AE}_{1,1,0}$ resulted in the lowest F-Measure of 0.47, while $Profile^{AE}_{1,0,1}$ resulted in the highest F-Measure

[1] A cross-lingual mapping problem is defined by the given ontologies being mono-lingual, but modelled using different natural languages.

κ	λ	μ	P^*	R^*	F^*
0	0	0	0.418	0.278	0.326
0	0	1	0.657	0.433	0.510
0	0	2	0.630	0.405	0.481
0	1	0	0.500	0.324	0.381
0	1	1	0.675	0.469	0.543
0	1	2	0.666	0,453	0.529
0	2	0	0.512	0.333	0.393
0	2	1	0.688	0.475	0.552
0	2	2	0.678	0.457	0.535
1	0	0	0.521	0.376	0.423
1	0	1	0.667	0.529	0.583
1	0	2	0.659	0.518	0.574
1	1	0	0.594	0.409	0.470
1	1	1	0.691	0.559	0.611
1	1	2	0.688	0.555	0.609
1	2	0	0.601	0.417	0.478
1	2	1	**0.699**	0.565	0.619
1	2	2	0.695	0.562	0.615
2	0	0	0.523	0.385	0.433
2	0	1	0.674	0.538	0.592
2	0	2	0.661	0.522	0.577
2	1	0	0.591	0.411	0.471
2	1	1	0.690	0.562	0.614
2	1	2	0.685	0.554	0.607
2	2	0	0.597	0.421	0.481
2	2	1	0.698	**0.570**	**0.622**
2	2	2	0.692	0.562	0.614

Table 7.2: Aggregated adapted Precision, Recall and F-Measure when evaluating all variations of our approach on the MultiFarm dataset.

of 0.583. We can also observe that combinations which include a variation of the RAD-set in the enriched profiles typically performed better than combinations that didn't.

Lastly, we can observe using all three types of description sets resulted in the highest measured F^* score. We can see that every combination of PAD, CAD and RAD resulted in an F^* score higher than 0.6. The best performing combination was $Profile^{AE}_{2,2,1}$ with an F^* score of 0.622. While this contrasts with the results of subsection 7.4.1, with $Profile^{AE}_{1,1,2}$ resulting in the highest score, we can observe that the difference in performed between $Profile^{AE}_{2,2,1}$ and $Profile^{AE}_{1,1,2}$ is not as distinct for this dataset.

Comparing RAD$_1$ with RAD$_2$ reveals a different trend than subsection 7.4.1.

Here, combinations which utilized RAD_1 performed slightly better than combinations which used RAD_2 instead. Taking the results of the benchmark dataset in mind, we thus cannot conclude whether RAD_1 or RAD_2 will result in a better performance in all cases. However, we can conclude that the inclusion of RAD_1 or RAD_2 in the concept profile does increase the performance of a profile similarity.

7.4.3 Influence of Partial Alignment Size

As detailed in section 7.3, the approach exploits the semantic relations of the provided partial alignment to enrich concept profiles with additional information. Hence, in order for a profile of a given concept x to be enriched, the partial alignment must contain a relation which specifies a concept which lies in the semantic neighbourhood of x. If this is not the case, then the profile of x will remain unchanged. It follows that, in order for this approach to be effective, the partial alignment must contain a sufficient amount of correspondences.

In this subsection we will investigate to what extent our approach is effective when supplied with partial alignments of varying size. To do this, we perform an evaluation similar to subsection 7.4.2. We evaluate our approach using the MultiFarm dataset by sampling random partial alignments from the reference 100 times for each task and aggregating the results. However, for a given combination $Profile^{AE}_{\kappa,\lambda,\mu}$ we perform this evaluation using different sizes of the partial alignment, with recall values $R(PA,R)$ ranging from 0.1 to 0.9 using increments of 0.1. The configurations $Profile^{AE}_{1,1,2}$ and $Profile^{AE}_{2,2,2}$ were utilized for this evaluation. The results of this experiment can be seen in Table 7.3.

		$R(PA,R)$								
		0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
$Profile^{AE}_{1,1,2}$	P^*	.436	.529	.587	.639	.688	.737	.788	.837	.883
	R^*	.272	.339	.406	.479	.556	.628	.711	.794	.891
	F^*	.324	.403	.471	.540	.609	.673	.744	.811	.883
$Profile^{AE}_{2,2,2}$	P^*	.432	.524	.587	.641	.692	.742	.790	.833	.859
	R^*	.277	.345	.411	.484	.561	.638	.721	.806	.904
	F^*	.328	.406	.475	.545	.614	.681	.751	.816	.878

Table 7.3: Aggregated adapted Precision, Recall and F-Measure on the MultiFarm dataset when varying the Recall of the supplied partial alignment.

The main trend which we can observe from Figure 7.3 is that $Profile^{AE}_{1,1,2}$ and $Profile^{AE}_{2,2,2}$ appear to be similarly affected by the size of PA. For every increment of the partial alignment size the F^* scores of $Profile^{AE}_{1,1,2}$ and $Profile^{AE}_{2,2,2}$ rise by roughly 0.07. For partial alignment recalls of 0.1 up to 0.8 we can observe that $Profile^{AE}_{2,2,2}$ performs slightly better than $Profile^{AE}_{1,1,2}$. While a different in F^* score by 0.005 can be dismissed as the result of the introduced variation of sampling the partial alignment, it is noteworthy here since it is consistent for all partial alignment recalls up to 0.8. Interestingly, for the highest recall value of *PA* $Profile^{AE}_{1,1,2}$ performed higher than $Profile^{AE}_{2,2,2}$. This is the result of $Profile^{AE}_{1,1,2}$ producing alignments with significantly higher precision at this configuration.

Overall, given the performances of $Profile^{AE}_{1,1,2}$ and $Profile^{AE}_{2,2,2}$, we can conclude that weighting descriptions of the anchored concepts such that the descriptions of closely related concepts are weighted higher results in a slight increase in performance for most recall levels of *PA*.

7.4.4 Comparison with Lexical Enrichment Systems

The main goal behind this work is to provide an approach that allows the enrichment of concept profile by exploiting the relations of a provided partial alignment. The reason behind this is that current enrichment methods exploit primarily lexical resources, which rely on the presence of an appropriate resource. In the previous sections we have established the performance of our approach using varying configurations, datasets, and partial alignment sizes. In this section, we will provide some interesting context for these results. Specifically, we aim to compare the results of our approach with the performances of matching systems tackling the same dataset while exploiting lexical resources. This allows us to establish whether an approach exploiting a partial alignment can produce alignments of similar quality than approaches exploiting lexical resources.

To do this, we will compare our approach to the performances of the OAEI participants on the MultiFarm dataset (Dragisic *et al.*, 2014). Here we will make the distinction between approaches utilizing no external resources, lexical resources and partial alignments. This allows us to see the benefit of exploiting a given type of external resource.

Furthermore, to provide an upper boundary for the potential performance on this dataset, we will also evaluate a method utilizing both lexical resources and partial alignments. To achieve this, we will re-evaluate the best performing configuration from sub-section 7.4.2. However, the profiles of this re-evaluation will be additionally enriched by translating the concept labels using the *Microsoft Bing* translator. This will provide an indication of how well a system may perform when utilizing both appropriate lexical resources and partial alignments. The comparison can be seen in Table 7.4.

From Table 7.4 we can make several observations. First, we can observe that every system utilizing either lexical resources or partial alignments performs significantly better than systems which do not. This is an expected result given the nature of this dataset. Of the system which do not exploit resources AOT has the highest performance with an F-Measure of 0.12.

Comparing the performance of $Profile^{AE}_{2,2,1}$ to the performance of system exploiting only lexical resources reveals an interesting observation. Specifically, we can see that the performance of these system is comparable. While the performances of LogMap and XMap were lower than $Profile^{AE}_{2,2,1}$, with an F-Measure of 0.62 the performance of AML is very close to the performance of $Profile^{AE}_{2,2,1}$. However, AML distinguishes itself from our approach by having a notably higher precision

Lex.	P. Align.	Matcher	Precision	Recall	F-Measure
yes	yes	$Profile^{AE}_{2,2,1}$ + Bing	*0.849*	*0.838*	*0.843*
yes	no	AML	0.95	0.48	0.62
yes	no	LogMap	0.94	0.27	0.41
yes	no	XMap	0.76	0.40	0.50
no	yes	$Profile^{AE}_{2,2,1}$	*0.698*	*0.570*	*0.622*
no	no	AOT	0.11	0.12	0.12
no	no	AOTL	0.27	0.01	0.02
no	no	LogMap-C	0.31	0.01	0.02
no	no	LogMapLt	0.25	0.01	0.02
no	no	MaasMatch	0.52	0.06	0.10
no	no	RSDLWB	0.34	0.01	0.02

Table 7.4: Comparison between the performance of our approach and the competitors of the 2014 OAEI competition on the MultiFarm dataset. Performances of approaches utilizing partial alignments are denoted in adapted precision, recall and F-Measure.

and a somewhat lower recall. In fact, all systems utilizing only lexical resources are characterized with a high precision, which implies that enriching ontologies using these resources only rarely leads to false-positive matches in terminology.

Lastly, we can observe the performance of our approach when paired with a lexical resource, specifically *Bing Translator*. Here the produced alignments reached an F^* score of 0.843, which is significantly higher than the OAEI participants. This implies that the correct correspondences which lexical-based systems identify differ significantly from the correct correspondences of a partial-alignment-based system. From this we can conclude that the two types of resources are complementary for matching problems with significant terminological gaps.

7.5 Chapter Conclusion and Future Work

We end this chapter by summarizing the results of the experiments (Subsection 7.5.1) and giving an outlook on future research (Subsection 7.5.2) based on the findings presented.

7.5.1 Chapter Conclusions

This chapter is aimed at answering *research question 4*, which concerns the matching of ontologies between which there exists a significant terminological gap by exploiting a given partial alignment. To answer this question, we developed an extension to an existing profile similarity, which interprets the semantic relations asserted by the partial alignment as additional exploitable relations. For the extended concept profile, we define three different sets of concepts, being the ancestors, descendants and otherwise related concepts. The extended profile is then created for each concept by identifying the anchors in each set and exploiting the descriptions of the anchored concepts.

First, we established the performance of our approach on the Benchmark dataset in sub-section 7.4.1 and on the MultiFarm dataset in sub-section 7.4.2. The evaluation on both datasets were performed with identical experimental set-ups. Every configuration of our approach is evaluated by repeatedly sampling a partial alignment from the reference alignments an aggregating the results of every evaluation. For both datasets we observed that our proposed extension significantly improves the performance of the profile similarity. We observed that exploring the relations to anchored parent-, child- and otherwise related concepts for a given concept all contribute to the quality of the output alignment. For both datasets, the top performance was measured in a configuration which exploited all three sets of exploitable concepts. In a further experiment we investigated the influence of weighting the added concept descriptions based on their distances to the given concept for partial alignment of varying sizes. We observed that weighting the descriptions based on their distance does resulted in a slightly better performance for most partial alignment sizes. However, for very large partial alignment sizes, the uniform weighting scheme performed better. Given that in a normal mapping scenario it is very unlikely that a given partial alignment exhibits such a significant size, we can conclude that for a real-world mapping scenario it is likely that a distance-based weighting scheme results in a better performance.

In the final experiment we compared the performance of our approach to the performances of other systems on the MultiFarm dataset. We established that the performance of our approach is comparable to the performances of systems exploiting lexical resources. Additionally, in order to give an indication of performance for a system exploiting both partial alignments and appropriate lexical resources, we executed our approach with an added enrichment system utilizing *Microsoft Bing.* This addition significantly improved the performance of our approach and resulted in a significantly higher performance than the compared OAEI systems. From this we can conclude that matching problems with significant terminological gaps can be matched with a high quality if both partial alignments and a lexical resource are available.

7.5.2 Future Research

We propose two directions of future research based on our finding presented in this chapter.

(1) In sub-section 7.4.4 we observed a performance indication of a mapping system utilizing both lexical resources and partial alignments. A future line of research would be the investigation to what extent this performance is replicable if a partial alignment is generated on-the-fly, instead of being given. This research should investigate which mapping technique produces reliable anchors for matching problems with significant terminological gaps and to what extent each technique impacts the subsequent matching performance.

(2) The core process of the proposed approach is the addition of anchored concept descriptions to an already existing profile. This includes terminology which might not re-occur in any concept description of either ontology. Future work could focus on the occurrence rates of terms within either ontologies. This could be in the form of filtering out terms which only occur once, or apply term weighting schemes such as TF-IDF.

Chapter 8

Conclusions and Future Research

This thesis investigated how auxiliary information can be used in order to enhance the performance of ontology mapping systems. This led to the formulation of our problem statement in Section 1.4.

> **Problem statement** *How can we improve ontology mapping systems by exploiting auxiliary information?*

To tackle the given problem statement, we focused the performed research on two types of auxiliary resources: lexical resources and partial alignments. Lexical resources present a research area with good potential due to their prevalent usage in existing matching systems, implying that there is a large group of benefactors of research in this area. Partial alignments pose a good target for research due to the limited amount of existing work regarding their exploitation. This implies that partial alignments are under-utilized in current matching systems, such that further research could enable potential performance gains for systems which do not utilize this resource yet. We have posed four research questions that need to be answered before addressing the problem statement.

In this chapter we will present the conclusions of this thesis. In Section 8.1 we will individually answer the four posed research questions. We formulate an answer to the problem statement in Section 8.2. Finally, we will present promising directions of future research in Section 8.3.

8.1 Conclusions on the Research Questions

The research questions stated in Chapter 1 concern the exploitation of auxiliary resources for ontology matching, i.e. (1) the disambiguation of concept senses for lexical similarities, (2) the exploitation of partial alignments in a general matching scenario, (3) the verification of partial alignment correspondences and (4) the exploitation of partial alignments for matching problems with significant terminological heterogeneities. They are dealt with in the following subsections, respectively.

8.1.1 Concept Disambiguation

Lexical resources are commonly utilized in ontology matching systems to derive similarity scores between individual ontology concepts. This is achieved in two core steps. First, for every concept the corresponding entries, also known as senses, within the given lexical resource need to be identified and associated with that concept. Second, the concepts are evaluated by comparing their associated senses. A metric which compares senses may utilize principles such as the semantic distance between senses in the resource or computing information theoretic commonalities. However, in order for such a metric to produce accurate results, the ontology concepts need to be associated with senses which accurately reflect their intended meaning. This issue has led us to the following research question.

> **Research question 1** *How can lexical sense definitions be accurately linked to ontology concepts?*

To answer this question, we proposed a virtual-document-based disambiguation method. This method utilizes an existing document model from an established profile similarity in order to generate virtual documents capable of representing both ontology concepts and lexical senses. This document model allows for of parametrized weighting of terms according to their respective origins within the ontology or lexical resource. For example, the terms originating from concept labels may receive higher weights than the concept annotation terms. Utilizing document similarity scores between the document of a given concept and the documents of potential lexical senses, a disambiguation policy is executed which only associates a sense to a concept, if its respective document fulfilled a given criterion.

Using several disambiguation policies, we evaluate the effect of our disambiguation framework on the *Conference* dataset by applying three different lexical similarities. Here we observe that the application of our disambiguation approach is beneficial for all tested lexical similarities lsm_1, lsm_2 and lsm_3. Further, we observe that for lsm_1 and lsm_3, stricter disambiguation policies produced better results, with the *MAX* policy resulting in the highest performance. For lsm_2 the disambiguation policy *A-Mean* resulted in the highest performance. In a further analysis, we compared the performance of our approach to the performances of existing matching systems. This has been achieved in two ways: (1) by entering a system based on our approach in the 2011 OAEI competition and (2) by evaluating our approach on a 2013 OAEI dataset and comparing the results to the performances of that year's participants. These comparisons revealed that a system utilizing our approach can perform competitively with state-of-the-art systems. This is especially noteworthy when taking into account the otherwise modest complexity of the tested system.

In a further experiment we quantified the potential benefit of utilizing the term weighting scheme of the document model and compared its performance to an information-retrieval-based weighting scheme, namely TF-IDF. For this, we generated two parameter sets by optimizing the weights on the *Benchmark* and *Con-*

ference datasets using Tree-Learning-Search. We observe that when trained on the appropriate dataset the weighting scheme of the document model produces better results at all recall levels. For recall levels of in the interval of $[0.4, 0.6]$ we observed the biggest differences in precision. In our last evaluation, we investigated the effect of the introduction of a disambiguation procedure has on the runtime. We observe an added runtime overhead of 3.65%. However, when also taking into account the gained efficiency due to the reduction of required calculations of sense similarities, we observe that addition of a disambiguation procedure improved the runtime of the lexical similarity. While the amount of runtime improvement depends on the efficiency of the used lexical similarity, we can conclude that the performance impact of a disambiguation procedure is small at most, whilst potentially being beneficial in the right circumstances.

8.1.2 Exploiting Partial Alignments

An additional type of available auxiliary resources are partial alignments. These are incomplete alignments stemming from previous matching efforts. An example of such an effort is a domain expert attempting to matching the given ontologies, but being unable to complete this task due to time constraints. In such a scenario, the core task is to identify the missing correspondences, such that the merger between the newly found correspondences and the given partial alignment can form a complete alignment. To do this, one can use the correspondences of the partial alignments, also referred to as anchors, in order to aide the matching process. This has lead us to the following research question.

> **Research question 2** *How can we exploit partial alignments in order to derive concept correspondences?*

With the intent of answering the second research question, we developed a technique which matches concepts by comparing their respective similarities towards the set of anchors. Concepts and anchors are compared by using a composite of different similarity metrics and their similarities are compiled into a profile for each concept, referred to as an anchor-profile. The core intuition behind this approach is that two concepts can be considered similar if they exhibit comparable degrees of similarity towards the given anchors.

To evaluate this approach, we introduced the measures of adapted Precision, Recall and F-Measure, referred to as P^*, R^* and F^* respectively. Unlike the measures from which these are derived, they take the presence of an input partial alignment PA into account. Thus, P^*, R^* and F^* accurately express the quality of the additionally computed correspondences with respect to their correctness, completeness and overall quality. We evaluated the approach on the *Benchmark* dataset by repeatedly randomly sampling the partial alignments from the reference alignment. This evaluation gave us a general performance indication, with an F^* score ranging in the interval of $[0.66, 0.78]$, whilst also showing that the matching performance is positively influenced by the size of the input partial alignment. An analysis on the performances on the different tracks reveals that the approach requires a composite similarity which utilizes various types of information from the ontologies in order to function in all conditions. A subsequent evaluation revealed that comparing a given ontology concept c with the anchor concept originating from the other ontology is preferable to comparing c with the anchor concept originating from the same ontology. Next, we systematically evaluated our approach using a spectrum of settings of the size and quality of the input partial alignment. This analysis revealed that while both these factors influence the performance of our approach, the quality of the partial alignment had a more severe impact on the performance than the size. From this we conclude that matching systems that generate partial alignments on-the-fly should focus their approach towards ensuring the correctness of the anchors. Lastly, we compared the quality of the computed correspondences to state-of-the-art system. This revealed that for larger partial alignments the quality is on par with the top systems for this dataset.

8.1.3 Filtering Partial Alignments

The results of answering the previous research question revealed that matching approaches utilizing partial alignments are influenced both by the size and correctness of the given partial alignments, with their correctness being a more influential factor. Hence, it is important that the partial alignment is evaluated such that incorrect correspondences can be discarded prior to matching. This led us to formulating the following research question.

> **Research question 3** *How can we evaluate whether partial alignment correspondences are reliable?*

To tackle the third research questions, we created a technique for the evaluation of anchors using feature selection techniques from the field of machine learning. We create a feature-space where every feature corresponds to a given anchor. We populate this feature space with labelled instances by generating reliable correspondences on-the-fly. An instance represents a series of consistency evaluations towards each anchor. The intuition behind this is that correct anchors are expected to produce predictable consistency evaluations for the instance set. We measure this predictability by applying a feature evaluation metric on each feature. We evaluate our approach on the *Conference* dataset by sampling the partial alignments, utilizing six different feature evaluation metrics and three different base similarities, which are used to compute the consistency scores for each instance. For each configuration, we compute the precision vs recall scores by ranking the anchors according to their evaluation scores. We compare these to three baseline rankings, which are created by ranking the anchors using purely the similarity scores of the base similarities. We observe that when utilizing a given base similarity, our approach can improve upon the performance of each corresponding base similarity. The most significant improvements were observed when utilizing a syntactic and lexical similarity. For the lexical similarity, the observed improvements were more significant at lower recall levels, with an increase of interpolated precision of up to 0.12. For the syntactic similarity the observed improvements were fairly consistent for all recall levels, with increases in interpolated precision typically falling in the interval of $[0.035, 0.057]$ above the baseline, resulting in a precision of 0.857 for the highest performing metric. For the lexical similarity, the baseline interpolated precision of approximately 0.821 for most recall levels was improved on by all tested metrics. The best performing metric resulted on a interpolated precision of 0.942, an improvement of 0.121. We observe that for base similarities for which significant improvements were observed, feature-evaluation-metrics utilizing class separability scores typically resulted in a better performance, particularly *Thornton's Separability Index.* We conclude that our approach presents a promising way of improving the utilization of existing similarity metrics for the evaluation of partial alignment correspondences.

8.1.4 Matching Terminologically Heterogeneous Ontologies

A challenging category of matching tasks are ontology pairs between which there exists no or an at most small terminological overlap. This causes many similarity metrics to produce unsatisfactory results. Typically, this issue is mitigated through the exploitation of lexical resources by enriching the ontologies with additional terminology. However, this approach imposes several issues, for example the availability of an appropriate resource for each problem. A way to circumvent these issues could be the exploitation of partial alignments instead of exploiting lexical resources. This led us to the following research question.

> **Research question 4** *To what extent can partial alignments be used in order to bridge a large terminological gap between ontologies?*

With the goal of answering research question 4, we developed an extension to a typical profile similarity. This extension interprets the semantic relations asserted in the partial alignment as additional relation types which the profile similarity can exploit. The core intuition is that natural languages often re-use concept terms when referring or defining more specific concepts. Thus, by extending a profile similarity through the exploitation of the added relations we can identify these re-occurrences of terminology and produce a better matching result. We refer to this extended profile as the anchor-enriched profile.

The evaluation of our approach requires a dataset consisting of matching tasks that are characterized by large terminological gaps. Hence, we evaluate our approach on a sub-set of the *Benchmark* dataset and the *MultiFarm* dataset. We repeatedly randomly sample the partial alignments from the reference alignment and compute an aggregate result for each evaluation. Further, the extension of the profile similarity is partitioned into three sets of descriptions, being the descriptions that have been gathered by exploring anchors that are ancestors, descendants or otherwise related concepts. We refer to these three sets as PAD, CAD and RAD respectively, with a subscript of 1 and 2 denoted whether the terms in the descriptions are weighted uniformly or proportionate to their semantic distance to the given concept. This allows us to see their individual effects on the mapping result and determine what configuration best suits the datasets. Our evaluation revealed that the addition of each description set benefited the performance of the profile similarity for both datasets. Additionally, for both datasets the highest performance was measured when utilizing all three types of description sets. However, while for the *Benchmark-subset* dataset the uniform weighting of terms produced better results, for the *MultiFarm* dataset the semantic-distance-based weighting was the preferable option. Overall, for the *Benchmark-subset* dataset we observed an improvement in F-Measure from 0.119 to 0.482 and for the *MultiFarm* dataset we observed an improvement in F-Measure from 0.326 to 0.622.

We further investigated the difference in performance between uniform and proportional weighting on the *MultiFarm* dataset by analysing the performances for different sizes of partial alignments. This comparison revealed that the difference in performance is consistent up to partial alignment recall values $R(PA, R)$ of 0.8. For $R(PA, R)$ values of 0.9 the uniform weighting method performed better. However, in real-world matching scenarios it is very unlikely that the given partial alignment exhibit a recall measure of 0.9. From this we conclude that for real-world matching cases a semantic-distance-based weighting scheme is preferable.

Finally, we compared the performance of our approach with with performances of other matching systems on the *MultiFarm* dataset. Here, we make the distinction between system utilizing no auxiliary resources, partial alignment or lexical resources. This comparison revealed that our approach performs significantly better than established matching systems utilizing no auxiliary resources and on par with AML, the top performing system utilizing an appropriate lexical resource. Furthermore, we re-evaluated our approach while also enriching the ontologies using an appropriate lexical resource, specifically *Microsoft Bing* translator. This provides a performance indication for a system utilizing both partial alignments and lexical resources. The resulting performance was characterized with an F-Measure of 0.843, a significant improvement compared to the performance of our approach without using *Bing* (F-Measure of 0.622) and the performance of AML (F-Measure of 0.62). From this we conclude that there is a significant performance potential for systems utilizing both types of auxiliary resources when faced with significant terminological gaps.

8.2 Conclusion to Problem Statement

After answering the four stated research questions, we are now able to provide an answer to the problem statement.

> **Problem statement** *How can we improve ontology mapping systems by exploiting auxiliary information?*

Taking the answers to the research questions into account we can see that there are numerous ways in which auxiliary information can be exploited to the benefit of ontology mapping systems. First, the utilization of lexical resources for the computation of semantic distances between concepts, where the accurate identification of concept senses can be achieved using a virtual document-based disambiguation process. Second, partial alignments can be exploited by creating a profile of similarities to the anchor concepts for each unmatched concept and comparing the resulting profiles. Third, our feature evaluation-based approach improves upon the performance of a normal similarity metric with regard to ensuring the correctness of the correspondences of input partial alignments. Fourth, the performance on mapping problems with significant terminological gaps can be improved upon by extending profile similarities such that they also exploit the asserted relations of a provided partial alignment.

8.3 Recommendations for Future Research

The research presented in this thesis indicates the following areas of interest for future research:

1. **Improving Concept Sense Disambiguation.** We identify three directions into which research can be performed to improve the performance of the disambiguation approach of Chapter 4:

   (a) The presented disambiguation method relies on the co-occurrences of exact terms for the determination of candidate senses and the sense similarity computation. Naming anomalies, such as spelling errors or non-standard syntax of compound words, can lead to significant issues. Next, after resolving naming anomalies the method needs to determine which senses best describe a given concept. For a sense to receive a higher similarity score, its annotation needs to contain exact terminology which also occurs in the annotation of the given concept. If the two annotations refer to the same entity using synonymous terms or a different syntax, then the result is that the similarity score of the ideal sense is not increased. Future research could aim to resolve these issues through the application of synonym extraction, spell-checking or soft document-similarity metrics.

   (b) An alternative to complement the shortcomings of our approach is by combining the results of multiple disambiguation techniques. This can be achieved by selecting the most appropriate disambiguation approach based on a heuristic evaluation or learning approach, or by combining the results of all given disambiguation approaches.

   (c) Methods of exploiting multiple source of lexical sources should be investigated. Instead of only utilizing the definitions provided by *WordNet*, one can also query other lexical sources such as *Wikipedia* or exploit the results of internet search engines such as *Google* or *Bing*.

2. **Improving Anchor Profiles.** The evaluation on the *Benchmark* dataset revealed that the robustness of the approach is influence by the choice of (compound-)similarity for the role as anchor similarity. Future research could be aimed at determining the best choice of similarities with regard to both matching quality in real-life matching cases and overall robustness. Additionally, the applicability of the approach can be widened by in researching methods of generating reliable anchors during runtime for matching problems that do not contain partial alignments.

3. **Improving Anchor Evaluation Techniques.** In Chapter 6 we described two core steps that are required for the filtering of possibly incorrect, the

anchor-evaluation step and the filter-policy step. The work of Chapter 6 presents an approach for the anchor-evaluation step, the result of which being a set of scores S. We propose that future work should be aimed at realizing the filter-policy step. Here, possible approaches need to be investigated which take as input the two ontologies O_1 and O_2, the partial alignment PA and the set of scores S, such that they produce a filtered partial alignment PA'. This would allow us to measure the actual benefit of a filtering approach by comparing the alignment quality of an anchor-based mapping approach before and after executing the filtering procedure. Additionally, we suggest addressing the robustness of the approach by testing future improvements on the *Benchmark* dataset.

4. **Improving Anchor-Enriched Profiles.** The performance indication of a system utilizing both lexical resources and partial alignments was created using the provided partial alignments of the experimental set-up. Future research should investigate whether this performance is replicable for systems which generate partial alignments during runtime. This would indicate whether the approach is applicable for terminological heterogeneous ontologies between which there does not exist a partial alignment.

References

Aas, Kjersti and Eikvil, Line (1999). Text categorisation: A survey. *Raport NR*, Vol. 941. [149]

Abeel, Thomas, Peer, Yves Van de, and Saeys, Yvan (2009). Java-ML: A machine learning library. *Journal of Machine Learning Research*, Vol. 10, pp. 931–934. [140]

Aguirre, José Luis, Cuenca Grau, Bernardo, Eckert, Kai, Euzenat, Jérôme, Ferrara, Alfio, Van Hague, Robert Willem, Hollink, Laura, Jimenez-Ruiz, Ernesto, Meilicke, Christian, Nikolov, Andriy, Ritze, Dominique, Scharffe, François, Shvaiko, Pavel, Sváb-Zamazal, Ondrej, Trojahn, Cássia, and Zapilko, Benjamin (2012). Results of the Ontology Alignment Evaluation Initiative 2012. *Proc. of the 7th ISWC workshop on ontology matching*, pp. 73–115. [47, 48, 73, 121, 129, 156]

Aldea, Arantza, López, Beatriz, Moreno, Antonio, Riaño, David, and Valls, Aïda (2001). A multi-agent system for organ transplant co-ordination. *Artificial intelligence in medicine*, pp. 413–416. Springer. [18]

Aleksovski, Zarko (2008). *Using background knowledge in ontology matching*. Ph.D. thesis, Vrije Universiteit Amsterdam. [21]

Androutsellis-Theotokis, Stephanos and Spinellis, Diomidis (2004). A survey of peer-to-peer content distribution technologies. *ACM Computing Surveys (CSUR)*, Vol. 36, No. 4, pp. 335–371. [10, 11]

Arens, Yigal, Knoblock, Craig A, and Shen, Wei-Min (1996). *Query reformulation for dynamic information integration*. Springer. [5, 8]

Aumueller, David, Do, Hong-Hai, Massmann, Sabine, and Rahm, Erhard (2005). Schema and ontology matching with COMA++. *Proceedings of the 2005 ACM SIGMOD international conference on Management of data*, pp. 906–908, ACM. [20, 63, 77, 116]

Banek, Marko, Vrdoljak, Boris, and Tjoa, A Min (2008). Word sense disambiguation as the primary step of ontology integration. *Database and Expert Systems Applications*, pp. 65–72, Springer. [94]

Banerjee, Satanjeev and Pedersen, Ted (2003). Extended gloss overlaps as a measure of semantic relatedness. *Proceedings of the 18th international joint conference on Artificial intelligence*, IJCAI'03, pp. 805–810, San Francisco, CA, USA. [90]

Bar-Hillel, Yehoshua (1960). The present status of automatic translation of languages. *Advances in computers*, Vol. 1, pp. 91–163. [88]

Batini, Carlo and Lenzerini, Maurizio (1984). A methodology for data schema integration in the entity relationship model. *Software Engineering, IEEE Transactions on*, Vol. SE-10, No. 6, pp. 650–664. [3]

Batini, Carlo, Lenzerini, Maurizio, and Navathe, Shamkant B. (1986). A comparative analysis of methodologies for database schema integration. *ACM computing surveys (CSUR)*, Vol. 18, No. 4, pp. 323–364. [3]

Bernstein, Philip A. and Rahm, Erhard (2000). Data warehouse scenarios for model management. *Conceptual Modeling—ER 2000*, pp. 1–15. Springer. [5]

Bodenreider, Olivier (2004). The unified medical language system (UMLS): integrating biomedical terminology. *Nucleic acids research*, Vol. 32, No. suppl 1, pp. D267–D270. [62, 86, 151]

Bollacker, Kurt, Evans, Colin, Paritosh, Praveen, Sturge, Tim, and Taylor, Jamie (2008). Freebase: a collaboratively created graph database for structuring human knowledge. *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pp. 1247–1250, ACM. [86]

Börner, Katy, Sanyal, Soma, and Vespignani, Alessandro (2007). Network science. *Annual review of information science and technology*, Vol. 41, No. 1, pp. 537–607. [12]

Bouquet, Paolo, Serafini, Luciano, and Zanobini, Stefano (2003). Semantic coordination: a new approach and an application. *The Semantic Web-ISWC 2003*, pp. 130–145. Springer. [5, 7]

Bouquet, Paolo, Serafini, Luciano, Zanobini, Stefano, and Sceffer, Simone (2006). Bootstrapping semantics on the web: meaning elicitation from schemas. *Proceedings of the 15th international conference on World Wide Web*, pp. 505–512, ACM. [64]

Brabham, Daren C. (2008). Crowdsourcing as a model for problem solving an introduction and cases. *Convergence: the international journal of research into new media technologies*, Vol. 14, No. 1, pp. 75–90. [23]

Broeck, Guy Van den and Driessens, Kurt (2011). Automatic discretization of actions and states in Monte-Carlo tree search. *Proceedings of the International Workshop on Machine Learning and Data Mining in and around Games (DMLG)*, pp. 1–12. [107]

Brunnermeier, Smita B. and Martin, Sheila A. (2002). Interoperability costs in the US automotive supply chain. *Supply Chain Management: An International Journal*, Vol. 7, No. 2, pp. 71–82. [203]

Budanitsky, Alexander and Hirst, Graeme (2001). Semantic distance in WordNet: An experimental, application-oriented evaluation of five measures. *Workshop on WordNet and other lexical resources, second meeting of the North American Chapter of the Association for Computational Linguistics*, pp. 29–34. [21, 85, 90, 96]

Budanitsky, Alexander and Hirst, Graeme (2006). Evaluating wordnet-based measures of lexical semantic relatedness. *Computational Linguistics*, Vol. 32, No. 1, pp. 13–47. [21]

Buitelaar, Paul, Cimiano, Philipp, Haase, Peter, and Sintek, Michael (2009). Towards Linguistically Grounded Ontologies. *The Semantic Web: Research and Applications*, Vol. 5554 of *Lecture Notes in Computer Science*, pp. 111–125. Springer Berlin / Heidelberg. ISBN 978–3–642–02120–6. [91]

Bunke, Horst (2000). Graph matching: Theoretical foundations, algorithms, and applications. *Proc. Vision Interface*, Vol. 2000, pp. 82–88. [61]

Bussler, Christoph, Fensel, Dieter, and Maedche, Alexander (2002). A conceptual architecture for semantic web enabled web services. *ACM Sigmod Record*, Vol. 31, No. 4, pp. 24–29. [13]

Calvanese, Diego, De Giacomo, Giuseppe, Lenzerini, Maurizio, Nardi, Daniele, and Rosati, Riccardo (1998). Information integration: Conceptual modeling and reasoning support. *Cooperative Information Systems, 1998. Proceedings. 3rd IFCIS International Conference on*, pp. 280–289, IEEE. [5, 8]

Cao, Bu-Qing, Li, Bing, and Xia, Qi-Ming (2009). A service-oriented qos-assured and multi-agent cloud computing architecture. *Cloud Computing*, pp. 644–649. Springer. [18]

Caraciolo, Caterina, Euzenat, Jérôme, Hollink, Laura, Ichise, Ryutaro, Isaac, Antoine, Malaisé, Véronique, Meilicke, Christian, Pane, Juan, Shvaiko, Pavel, Stuckenschmidt, Heiner, et al. (2008). Results of the ontology alignment evaluation initiative 2008. *Proc. 3rd ISWC workshop on ontology matching (OM)*, pp. 73–119, No commercial editor. [45]

Chavez, Anthony, Moukas, Alexandros, and Maes, Pattie (1997). Challenger: A multi-agent system for distributed resource allocation. *Proceedings of the first international conference on Autonomous agents*, pp. 323–331, ACM. [18]

Cheatham, Michelle (2011). MapSSS Results for OAEI 2011. *Proceedings of The Sixth ISWC International Workshop on Ontology Matching(OM).* [76]

Cheatham, Michelle (2013). StringsAuto and MapSSS results for OAEI 2013. *Proceedings of The Eighth ISWC International Workshop on Ontology Matching(OM).* [71, 76]

Chen, Siqi and Weiss, Gerhard (2012). An Efficient and Adaptive Approach to Negotiation in Complex Environments. *ECAI'2012*, pp. 228–233, IOS Press. [18]

Coalition, DAML-S, Ankolekar, Anupriya, Burstein, Mark, Hobbs, Jerry R., Lassila, Ora, Martin, David, McDermott, Drew, McIlraith, Sheila A., Narayanan, Srini, and Paolucci, Massimo (2002). DAML-S: Web service description for the semantic Web. *The Semantic Web-ISWC*, pp. 348–363, Springer. [14]

Cohen, William, Ravikumar, Pradeep, and Fienberg, Stephen (2003). A comparison of string metrics for matching names and records. *KDD Workshop on Data Cleaning and Object Consolidation*, Vol. 3, pp. 73–78. [58]

Croft, W Bruce, Metzler, Donald, and Strohman, Trevor (2009). *Search Engines: Information Retrieval in Practice*. Addison-Wesley Publishing Company, USA, 1st edition. ISBN 0136072240, 9780136072249. [100]

Cruz, Isabel F., Sunna, William, and Chaudhry, Anjli (2004). Semi-automatic ontology alignment for geospatial data integration. *Geographic Information Science*, pp. 51–66. Springer. [25]

Cruz, Isabel F, Antonelli, Flavio Palandri, and Stroe, Cosmin (2009). AgreementMaker: efficient matching for large real-world schemas and ontologies. *Proc. VLDB Endow.*, Vol. 2, No. 2, pp. 1586–1589. ISSN 2150–8097. [70, 73, 148, 149]

Cruz, Isabel F., Fabiani, Alessio, Caimi, Federico, Stroe, Cosmin, and Palmonari, Matteo (2012). Automatic configuration selection using ontology matching task profiling. *The Semantic Web: Research and Applications*, pp. 179–194. Springer. [21]

Cruz, Isabel F., Palmonari, Matteo, Caimi, Federico, and Stroe, Cosmin (2013). Building linked ontologies with high precision using subclass mapping discovery. *Artificial Intelligence Review*, Vol. 40, No. 2, pp. 127–145. [74, 86, 87, 91]

d'Aquin, Mathieu and Lewen, Holger (2009). Cupboard–a place to expose your ontologies to applications and the community. *The Semantic Web: Research and Applications*, pp. 913–918. Springer. [24]

De Melo, Gerard and Weikum, Gerhard (2009). Towards a universal wordnet by learning from combined evidence. *Proceedings of the 18th ACM conference on Information and knowledge management*, pp. 513–522, ACM. [86]

Ding, Ying, Korotkiy, M, Omelayenko, Borys, Kartseva, V, Zykov, V, Klein, Michel, Schulten, Ellen, and Fensel, Dieter (2002). Goldenbullet: Automated classification of product data in e-commerce. *Proceedings of the 5th International Conference on Business Information Systems.* [5, 7]

Dodge, Yadolah (2008). Pooled Variance. *The Concise Encyclopedia of Statistics*, pp. 427–428. Springer New York. ISBN 978–0–387–31742–7. [121]

Do, Hong-Hai and Rahm, Erhard (2002). COMA: a system for flexible combination of schema matching approaches. *Proceedings of the 28th international conference on Very Large Data Bases*, pp. 610–621, VLDB Endowment. [20, 69, 77]

Dragisic, Zlatan, Eckert, Kai, Euzenat, Jérôme, Faria, Daniel, Ferrara, Alfio, Granada, Roger, Ivanova, Valentina, Jimenez-Ruiz, Ernesto, Kempf, Andreas, Lambrix, Patrick, et al. (2014). Results of theOntology Alignment Evaluation Initiative 2014. *International Workshop on Ontology Matching*, pp. 61–104. [157, 159, 162]

Duda, Richard O., Hart, Peter E., and Stork, David G. (1999). *Pattern classification*. John Wiley & Sons,. [66]

Ehrig, Marc (2006). *Ontology alignment: bridging the semantic gap*, Vol. 4. Springer. [31, 58]

Ehrig, Marc and Euzenat, Jérôme (2005). Relaxed precision and recall for ontology matching. *Proc. K-Cap 2005 workshop on Integrating ontology*, pp. 25–32. [42, 43, 44]

Ehrig, Marc and Staab, Steffen (2004). QOM–quick ontology mapping. *The Semantic Web–ISWC 2004*, pp. 683–697. Springer. [20, 69, 76]

Ehrig, Marc and Sure, York (2004). Ontology mapping–an integrated approach. *The Semantic Web: Research and Applications*, pp. 76–91. Springer. [67, 69, 76, 84]

Ehrig, Marc, Schmitz, Christoph, Staab, Steffen, Tane, Julien, and Tempich, Christoph (2004). Towards evaluation of peer-to-peer-based distributed knowledge management systems. *Agent-Mediated Knowledge Management*, pp. 73–88. Springer. [10]

Elfeky, Mohamed G., Verykios, Vassilios S., and Elmagarmid, Ahmed K. (2002). TAILOR: A record linkage toolbox. *Data Engineering, 2002. Proceedings. 18th International Conference on*, pp. 17–28, IEEE. [60]

Euzenat, Jérôme (2001). Towards a principled approach to semantic interoperability. *Proceedings of the IJCAI-01 Workshop on Ontologies and Information Sharing*, pp. 19–25. [27]

Euzenat, Jérôme (2004a). An API for ontology alignment. *The Semantic Web–ISWC 2004*, pp. 698–712. Springer. [24]

Euzenat, Jérôme (2004b). Introduction to the EON Ontology alignment contest. *Proc. 3rd ISWC2004 workshop on Evaluation of Ontology-based tools (EON)*, pp. 47–50. [46]

Euzenat, Jérôme (2005). Alignment infrastructure for ontology mediation and other applications. *Proc. 1st ICSOC international workshop on Mediation in semantic web services*, pp. 81–95. [23, 24]

Euzenat, Jérôme and Shvaiko, Pavel (2007). *Ontology Matching*, Vol. 18. Springer Berlin. [31, 56, 57, 58, 61, 62, 64, 65, 69, 73, 87, 149]

Euzenat, Jérôme, Stuckenschmidt, Heiner, Yatskevich, Mikalai, et al. (2005). Introduction to the ontology alignment evaluation 2005. *Proc. K-Cap 2005 workshop on Integrating ontology*, pp. 61–71. [46]

Euzenat, Jérôme, Ferrara, Alfio, Hollink, Laura, Isaac, Antoine, Joslyn, Cliff, Malaisé, Véronique, Meilicke, Christian, Nikolov, Andriy, Pane, Juan, Sabou, Marta, et al. (2009a). Results of the ontology alignment evaluation initiative 2009. *Proc. 4th ISWC workshop on ontology matching (OM)*, pp. 73–126. [48]

Euzenat, Jérôme, Ferrara, Alfio, Hollink, Laura, Isaac, Antoine, Joslyn, Cliff, Malaisé, Véronique, Meilicke, Christian, Nikolov, Andriy, Pane, Juan, Sabou, Marta, et al. (2009b). Results of the ontology alignment evaluation initiative 2009. *Proc. 4th ISWC workshop on ontology matching (OM)*, pp. 73–126. [49]

Euzenat, J., Ferrara, A., Meilicke, C., Pane, J., Scharffe, F., Shvaiko, P., Stuckenschmidt, H., Svab-Zamazal, O., Svatek, V., and Trojahn, C. (2010). First Results of the Ontology Alignment Evaluation Initiative 2010. *Proceedings of ISWC Workshop on OM*, pp. 85–117. [20, 39, 49]

Euzenat, Jérôme, Ferrara, Alfio, Hague, Robert Willem van, Hollink, Laura, Meilicke, Christian, Nikolov, Andriy, Scharffe, Francois, Shvaiko, Pavel, Stuckenschmidt, Heiner, Svab-Zamazal, Ondrej, and Santo, Cassia Trojahn dos (2011a). Results of the ontology alignment evaluation initiative 2011. *Proc. 6th ISWC workshop on ontology matching (OM)*, pp. 85–110. [49]

Euzenat, Jérôme, Meilicke, Christian, Stuckenschmidt, Heiner, Shvaiko, Pavel, and Trojahn, Cássia (2011b). Ontology alignment evaluation initiative: Six years of experience. *Journal on data semantics XV*, pp. 158–192. Springer. [20, 46, 101]

Euzenat, J., Ferrara, A., Hague, R.W. van, Hollink, L., Meilicke, C., Nikolov, A., Scharffe, F., Shvaiko, P., Stuckenschmidt, H., Svab-Zamazal, O., and Santos, C. Trojahn dos (2011c). Results of the Ontology Alignment Evaluation Initiative 2011. *Proc. 6th ISWC workshop on ontology matching (OM), Bonn (DE)*, pp. 85–110. [101, 103, 105]

Falconer, Sean M and Storey, Margaret-Anne (2007). A cognitive support framework for ontology mapping. *The Semantic Web*, pp. 114–127. Springer. [22]

Fan, Wenfei, Li, Jianzhong, Ma, Shuai, Wang, Hongzhi, and Wu, Yinghui (2010). Graph homomorphism revisited for graph matching. *Proceedings of the VLDB Endowment*, Vol. 3, Nos. 1–2, pp. 1161–1172. [61]

Faria, Daniel, Pesquita, Catia, Santos, Emanuel, Palmonari, Matteo, Cruz, Isabel F, and Couto, Francisco M (2013). The agreementmakerlight ontology matching system. *On the Move to Meaningful Internet Systems: OTM 2013 Conferences*, pp. 527–541, Springer. [20, 73]

Ferrara, Alfio, Lorusso, Davide, Montanelli, Stefano, and Varese, Gaia (2008). Towards a benchmark for instance matching. *The 7th International Semantic Web Conference*, pp. 37–48. [14]

Ferrucci, David, Brown, Eric, Chu-Carroll, Jennifer, Fan, James, Gondek, David, Kalyanpur, Aditya A, Lally, Adam, Murdock, J William, Nyberg, Eric, Prager, John, et al. (2010). Building Watson: An overview of the DeepQA project. *AI magazine*, Vol. 31, No. 3, pp. 59–79. [17]

Finin, Tim, Fritzson, Richard, McKay, Don, and McEntire, Robin (1994). KQML as an agent communication language. *Proceedings of the third international conference on Information and knowledge management*, pp. 456–463, ACM. [18]

FIPA, TCC (2008). Fipa communicative act library specification. *Foundation for Intelligent Physical Agents, http://www. fipa. org/specs/fipa00037/SC00037J. html (30.6. 2004).* [18]

Fisher, Ronald A. (1936). The use of multiple measurements in taxinomic problems. *Annals of Eugenics*, Vol. 7, No. 2, pp. 179–188. ISSN 2050–1439. [140]

Flannery, Brian P., Press, William H., Teukolsky, Saul A., and Vetterling, William (1992). Numerical recipes in C. *Press Syndicate of the University of Cambridge, New York.* [139]

Gale, David and Shapley, Lloyd S. (1962). College Admissions and the Stability of Marriage. *American Mathematical Monthly*, Vol. 69, No. 1, pp. 9–15. [71]

Gale, William A., Church, Kenneth W., and Yarowsky, David (1992). A Method for Disambiguating Word Senses in a Large Corpus. *Computers and the Humanities*, Vol. 26, No. 5/6, pp. pp. 415–439. [88]

Gallaher, Michael P., O'Connor, Alan C., Dettbarn, John L., and Gilday, Linda T. (2004). Cost analysis of inadequate interoperability in the US capital facilities industry. *National Institute of Standards and Technology (NIST).* [202]

Gangemi, Aldo, Guarino, Nicola, Masolo, Claudio, and Oltramari, Alessandro (2003). Sweetening wordnet with dolce. *AI magazine*, Vol. 24, No. 3, p. 13. [62]

Gao, Jian-Bo, Zhang, Bao-Wen, and Chen, Xiao-Hua (2015). A WordNet-based semantic similarity measurement combining edge-counting and information content theory. *Engineering Applications of Artificial Intelligence*, Vol. 39, pp. 80–88. [95]

Giunchiglia, Fausto and Shvaiko, Pavel (2003). Semantic matching. *The Knowledge Engineering Review*, Vol. 18, No. 03, pp. 265–280. [64]

Giunchiglia, Fausto and Yatskevich, Mikalai (2004). Element Level Semantic Matching. *Meaning Coordination and Negotiation workshop (MCN-04), collocated at ISWC-2004.* [85]

Giunchiglia, Fausto, Shvaiko, Pavel, and Yatskevich, Mikalai (2004). S-Match: an algorithm and an implementation of semantic matching. *ESWS*, Vol. 3053, pp. 61–75, Springer. [74]

Giunchiglia, Fausto, Yatskevich, Mikalai, and Shvaiko, Pavel (2007). Semantic matching: Algorithms and implementation. *Journal on Data Semantics IX*, pp. 1–38. Springer. [19]

Giunchiglia, Fausto, Yatskevich, Mikalai, Avesani, Paolo, and Shivaiko, Pavel (2009). A large dataset for the evaluation of ontology matching. *The Knowledge Engineering Review*, Vol. 24, No. 02, pp. 137–157. [38]

Gligorov, Risto, Kate, Warner ten, Aleksovski, Zharko, and Harmelen, Frank van (2007). Using Google distance to weight approximate ontology matches. *Proceedings of the 16th international conference on World Wide Web*, pp. 767–776, ACM. [20]

Grau, Bernardo Cuenca, Dragisic, Zlatan, Eckert, Kai, Euzenat, Jérôme, Ferrara, Alfio, Granada, Roger, Ivanova, Valentina, Jiménez-Ruiz, Ernesto, Kempf, Andreas Oskar, Lambrix, Patrick, et al. (2013). Results of the Ontology Alignment Evaluation Initiative 2013. *Proc. 8th ISWC workshop on ontology matching (OM)*, pp. 61–100. [20, 21, 46, 47, 48, 50, 73, 105, 140]

Gross, Anika, Hartung, Michael, Kirsten, Toralf, and Rahm, Erhard (2012). GOMMA results for OAEI 2012. *Ontology Matching Workshop. International Semantic Web Conference.* [20]

Gruber, Thomas R. (1993). A translation approach to portable ontology specifications. *Knowledge acquisition*, Vol. 5, No. 2, pp. 199–220. [1]

Gulić, Marko and Vrdoljak, Boris (2013). CroMatcher-Results for OAEI 2013. *Proceedings of The Eigth ISWC International Workshop on Ontology Matching*, pp. 117–122. [72, 77, 148]

Guyon, Isabelle and Elisseeff, André (2003). An introduction to variable and feature selection. *Journal of Machine Learning Research*, Vol. 3, pp. 1157–1182. [138]

Halevy, Alon Y., Ashish, Naveen, Bitton, Dina, Carey, Michael, Draper, Denise, Pollock, Jeff, Rosenthal, Arnon, and Sikka, Vishal (2005). Enterprise information integration: successes, challenges and controversies. *Proceedings of the 2005 ACM SIGMOD international conference on Management of data*, pp. 778–787, ACM. [5, 8]

Halevy, Alon, Rajaraman, Anand, and Ordille, Joann (2006). Data integration: the teenage years. *Proceedings of the 32nd international conference on Very large data bases*, pp. 9–16, VLDB Endowment. [5, 8]

Hamming, Richard W. (1950). Error detecting and error correcting codes. *Bell System technical journal*, Vol. 29, No. 2, pp. 147–160. [60]

Hepp, Martin and Roman, Dumitru (2007). An Ontology Framework for Semantic Business Process Management. *Wirtschaftsinformatik (1)*, pp. 423–440. [10]

Hepp, Martin, Bachlechner, Daniel, and Siorpaes, Katharina (2006). OntoWiki: community-driven ontology engineering and ontology usage based on Wikis. *Proceedings of the 2006 international symposium on Wikis*, pp. 143–144, ACM. [8]

Hepp, Martin, Leukel, Joerg, and Schmitz, Volker (2007). A quantitative analysis of product categorization standards: content, coverage, and maintenance of eCl@ ss, UNSPSC, eOTD, and the RosettaNet Technical Dictionary. *Knowledge and Information Systems*, Vol. 13, No. 1, pp. 77–114. [8]

Hertling, Sven and Paulheim, Heiko (2012a). WikiMatch–Using Wikipedia for Ontology Matching. *Proceedings of The Seventh ISWC International Workshop on Ontology Matching(OM)*, p. 37, Citeseer. [77]

Hertling, Sven and Paulheim, Heiko (2012b). WikiMatch Results for OEAI 2012. *Proceedings of The Seventh ISWC International Workshop on Ontology Matching(OM)*, pp. 220–225. [77]

Hindle, Donald and Rooth, Mats (1993). Structural ambiguity and lexical relations. *Computational linguistics*, Vol. 19, No. 1, pp. 103–120. [90]

Hsu, Feng-Hsiung (2002). *Behind Deep Blue: Building the computer that defeated the world chess champion.* Princeton University Press. [17]

Hu, Wei and Qu, Yuzhong (2008). Falcon-AO: A practical ontology matching system. *Web Semantics: Science, Services and Agents on the World Wide Web*, Vol. 6, No. 3, pp. 237–239. [75]

Ichise, Ryutaro (2008). Machine learning approach for ontology mapping using multiple concept similarity measures. *Proceedings of the Seventh IEEE/ACIS International Conference on Computer and Information Science(ICIS 08)*, pp. 340–346, IEEE. [66]

Ide, Nancy and Véronis, Jean (1998). Introduction to the special issue on word sense disambiguation: the state of the art. *Computational linguistics*, Vol. 24, No. 1, pp. 2–40. [88]

Isaac, Antoine, Wang, Shenghui, Zinn, Claus, Matthezing, Henk, Meij, Lourens van der, and Schlobach, Stefan (2009). Evaluating thesaurus alignments for semantic interoperability in the library domain. *IEEE Intelligent Systems*, Vol. 24, No. 2, pp. 76–86. [48]

Ives, Zachary G., Halevy, Alon Y., Mork, Peter, and Tatarinov, Igor (2004). Piazza: mediation and integration infrastructure for semantic web data. *Web Semantics: Science, Services and Agents on the World Wide Web*, Vol. 1, No. 2, pp. 155–175. [11]

Jaccard, Paul (1901). Étude comparative de la distribution florale dans une portion des Alpes et des Jura. *Bulletin del la Société Vaudoise des Sciences Naturelles*, Vol. 37, pp. 547–579. [60]

Jain, Prateek, Hitzler, Pascal, Sheth, Amit P, Verma, Kunal, and Yeh, Peter Z (2010). Ontology alignment for linked open data. *The Semantic Web–ISWC 2010*, pp. 402–417. Springer. [20]

Jaro, Matthew A. (1989). Advances in Record-Linkage Methodology as Applied to Matching the 1985 Census of Tampa, Florida. *Journal of the American Statistical Association*, Vol. 84, No. 406, pp. 414–420. ISSN 01621459. [101, 141]

Jean-Mary, Yves R., Shironoshita, E. Patrick, and Kabuka, Mansur R. (2009). Ontology matching with semantic verification. *Web Semant.*, Vol. 7, pp. 235–251. ISSN 1570–8268. [74]

Jian, Ningsheng, Hu, Wei, Cheng, Gong, and Qu, Yuzhong (2005). Falcon-ao: Aligning ontologies with falcon. *Proceedings of K-CAP Workshop on Integrating Ontologies*, pp. 85–91. [75]

Jiménez-Ruiz, Ernesto and Cuenca Grau, Bernardo (2011). LogMap: logic-based and scalable ontology matching. *The Semantic Web–International Semantic Web Conference (ISWC)*, pp. 273–288, Springer Berlin/Heidelberg. [75, 116, 135, 149]

Jiménez-Ruiz, Ernesto, Cuenca Grau, Bernardo, and Horrocks, Ian (2012a). LogMap and LogMapLt Results for OAEI 2012. *7th International Workshop on Ontology Matching (OM).* [20]

Jiménez-Ruiz, Ernesto, Grau, Bernardo Cuenca, Zhou, Yujiao, and Horrocks, Ian (2012b). Large-scale Interactive Ontology Matching: Algorithms and Implementation. *ECAI*, Vol. 242, pp. 444–449. [75, 116, 135]

Kalfoglou, Yannis and Schorlemmer, Marco (2003). Ontology mapping: the state of the art. *The knowledge engineering review*, Vol. 18, No. 1, pp. 1–31. [86]

Kang, Jaewoo and Naughton, Jeffrey F (2003). On schema matching with opaque column names and data values. *Proceedings of the 2003 ACM SIGMOD international conference on Management of data*, pp. 205–216, ACM. [58]

Khiat, Abderrahmane and Benaissa, Moussa (2014). AOT/AOTL Results for OAEI 2014. *Proceedings of The Ninth ISWC International Workshop on Ontology Matching(OM).* [76]

Killeen, Peter R (2005). An alternative to null-hypothesis significance tests. *Psychological science*, Vol. 16, No. 5, pp. 345–353. [121]

Kim, Won and Seo, Jungyun (1991). Classifying schematic and data heterogeneity in multidatabase systems. *Computer*, Vol. 24, No. 12, pp. 12–18. [3]

Klein, Michel and Noy, Natalya F. (2003). A component-based framework for ontology evolution. *Proceedings of the IJCAI*, Vol. 3, Citeseer. [10]

Klusch, Matthias, Fries, Benedikt, and Sycara, Katia (2006). Automated semantic web service discovery with OWLS-MX. *Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems*, pp. 915–922, ACM. [14]

Klusch, Matthias, Fries, Benedikt, and Sycara, Katia (2009). OWLS-MX: A hybrid Semantic Web service matchmaker for OWL-S services. *Web Semantics: Science, Services and Agents on the World Wide Web*, Vol. 7, No. 2, pp. 121–133. [13]

Kotis, Konstantinos, Valarakos, Alexandros G., and Vouros, George A. (2006a). AUTOMS: Automated Ontology Mapping through Synthesis of methods. *Proceedings of Ontology Matching (OM)*, pp. 96–106. [75, 91]

Kotis, Konstantinos, Vouros, George A, and Stergiou, Konstantinos (2006b). Towards automatic merging of domain ontologies: The HCONE-merge approach. *Web Semantics: Science, Services and Agents on the World Wide Web*, Vol. 4, No. 1, pp. 60–79. ISSN 1570–8268. [75, 91]

Kuhn, Harold W. (1955). The Hungarian method for the assignment problem. *Naval research logistics quarterly*, Vol. 2, Nos. 1–2, pp. 83–97. [70]

Labrou, Yannis, Finin, Tim, and Peng, Yun (1999). Agent communication languages: The current landscape. *IEEE Intelligent systems*, Vol. 14, No. 2, pp. 45–52. [18]

Lambrix, Patrick and Liu, Qiang (2009). Using partial reference alignments to align ontologies. *The Semantic Web: Research and Applications*, pp. 188–202. Springer. [20]

Lassila, Ora, Swick, Ralph R., and W3C (1998). Resource Description Framework (RDF) Model and Syntax Specification. [89]

Lee, Mong Li, Yang, Liang Huai, Hsu, Wynne, and Yang, Xia (2002). XClust: clustering XML schemas for effective integration. *Proceedings of the eleventh international conference on Information and knowledge management*, pp. 292–299, ACM. [60]

Lee, Yoonkyong, Sayyadian, Mayssam, Doan, AnHai, and Rosenthal, Arnon S (2007). eTuner: tuning schema matching software using synthetic scenarios. *The VLDB Journal—The International Journal on Very Large Data Bases*, Vol. 16, No. 1, pp. 97–122. [21]

Lei, Yuangui, Uren, Victoria, and Motta, Enrico (2006). Semsearch: A search engine for the semantic web. *Managing Knowledge in a World of Networks*, pp. 238–245. Springer. [16]

Lenzerini, Maurizio (2002). Data integration: A theoretical perspective. *Proceedings of the twenty-first ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pp. 233–246, ACM. [5, 9]

Lesk, Michael (1986). Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an ice cream cone. *Proceedings of the 5th annual international conference on Systems documentation*, SIGDOC '86, pp. 24–26. ISBN 0–89791–224–1. [90]

Levi, Giorgio (1973). A note on the derivation of maximal common subgraphs of two directed or undirected graphs. *Calcolo*, Vol. 9, No. 4, pp. 341–352. [61]

Lindberg, Donald A., Humphreys, Betsy L., and McCray, Alexa T. (1993). The Unified Medical Language System. *Methods of information in medicine*, Vol. 32, No. 4, pp. 281–291. [49]

Litkowski, Kenneth C. (1997). Desiderata for tagging with WordNet synsets or MCCA categories. *fourth meeting of the ACL Special Interest Group on the Lexicon. Washington, DC: Association for Computational Linguistics.* [88]

Li, Juanzi, Tang, Jie, Li, Yi, and Luo, Qiong (2009). Rimom: A dynamic multistrategy ontology alignment framework. *Knowledge and Data Engineering, IEEE Transactions on*, Vol. 21, No. 8, pp. 1218–1232. [76, 148, 149]

Locke, William Nash and Booth, Andrew Donald (1955). *Machine translation of languages: fourteen essays.* Published jointly by Technology Press of the Massachusetts Institute of Technology and Wiley, New York. [88]

Lopez, Vanessa, Pasin, Michele, and Motta, Enrico (2005). Aqualog: An ontology-portable question answering system for the semantic web. *The Semantic Web: Research and Applications*, pp. 546–562. Springer. [16]

Lopez, Vanessa, Uren, Victoria, Motta, Enrico, and Pasin, Michele (2007). AquaLog: An ontology-driven question answering system for organizational semantic intranets. *Web Semantics: Science, Services and Agents on the World Wide Web*, Vol. 5, No. 2, pp. 72–105. [16]

Manning, Christopher D., Raghavan, Prabhakar, and Schütze, Hinrich (2008). *Introduction to information retrieval*, Vol. 1. Cambridge university press Cambridge. [62, 149]

Mao, Ming and Peng, Yefei (2006). PRIOR System: Results for OAEI 2006. *Ontology Matching*, p. 173. [72, 75]

Mao, Ming, Peng, Yefei, and Spring, Michael (2007). A profile propagation and information retrieval based ontology mapping approach. *Proceedings of the Third International Conference on Semantics, Knowledge and Grid*, pp. 164–169, IEEE. [62, 74, 117, 148, 150, 154]

Maree, Mohammed and Belkhatir, Mohammed (2014). Addressing semantic heterogeneity through multiple knowledge base assisted merging of domain-specific ontologies. *Knowledge-Based Systems*, Vol. 73, No. 0, pp. 199 – 211. [91]

Marshall, Ian (1983). Choice of Grammatical Word-Class without Global Syntactic Analysis: Tagging Words in the LOB Corpus. *Computers and the Humanities*, Vol. 17, No. 3, pp. pp. 139–150. ISSN 00104817. [89]

Martin, David, Burstein, Mark, Hobbs, Jerry, Lassila, Ora, McDermott, Drew, McIlraith, Sheila, Narayanan, Srini, Paolucci, Massimo, Parsia, Bijan, Payne, Terry, et al. (2004). OWL-S: Semantic markup for web services. *W3C member submission*, Vol. 22, pp. 2007–04. [14]

Matuszek, Cynthia, Cabral, John, Witbrock, Michael, and DeOliveira, John (2006). An Introduction to the Syntax and Content of Cyc. *Proceedings of the 2006 AAAI Spring Symposium on Formalizing and Compiling Background Knowledge and Its Applications to Knowledge Representation and Question Answering*, pp. 44–49. [21, 62, 86]

Maximilien, E Michael and Singh, Munindar P (2004). Toward autonomic web services trust and selection. *Proceedings of the 2nd international conference on Service oriented computing*, pp. 212–221, ACM. [13]

McCann, Robert, Shen, Warren, and Doan, AnHai (2008). Matching schemas in online communities: A web 2.0 approach. *Data Engineering, 2008. ICDE 2008. IEEE 24th International Conference on*, pp. 110–119, IEEE. [23]

McCarthy, Diana, Koeling, Rob, Weeds, Julie, and Carroll, John (2004). Finding predominant word senses in untagged text. *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, p. 279, Association for Computational Linguistics. [87]

McCord, Michael C., Murdock, J. William, and Boguraev, Branimir K. (2012). Deep parsing in Watson. *IBM Journal of Research and Development*, Vol. 56, No. 3.4, pp. 3–1. [17]

McCrae, John, Spohr, Dennis, and Cimiano, Philipp (2011). Linking Lexical Resources and Ontologies on the Semantic Web with Lemon. *The Semantic Web: Research and Applications*, Vol. 6643 of *Lecture Notes in Computer Science*, pp. 245–259. Springer. ISBN 978–3–642–21033–4. [91]

McGuinness, Deborah L. and Van Harmelen, Frank (2004). OWL Web Ontology Language Overview. W3C recommendation, W3C. [89]

Medjahed, Brahim, Bouguettaya, Athman, and Elmagarmid, Ahmed K (2003). Composing web services on the semantic web. *The VLDB Journal*, Vol. 12, No. 4, pp. 333–351. [13, 14]

Meilicke, Christian and Stuckenschmidt, Heiner (2007). Analyzing mapping extraction approaches. *Proceedings of the ISWC 2007 Workshop on Ontology Matching*, pp. 25–36. [71, 72, 101]

Meilicke, Christian, García-Castro, Raúl, Freitas, Fred, Van Hage, Willem Robert, Montiel-Ponsoda, Elena, Azevedo, Ryan Ribeiro de, Stuckenschmidt, Heiner, Šváb-Zamazal, Ondřej, Svátek, Vojtěch, Tamilin, Andrei, et al. (2012). MultiFarm: A benchmark for multilingual ontology matching. *Web Semantics: Science, Services and Agents on the World Wide Web*, Vol. 15, pp. 62–68. [49, 156]

Melnik, Sergey, Garcia-Molina, Hector, and Rahm, Erhard (2002). Similarity flooding: A versatile graph matching algorithm and its application to schema matching. *Data Engineering, 2002. Proceedings. 18th International Conference on*, pp. 117–128, IEEE. [69, 76]

Mihalcea, Rada (2006). Knowledge-based methods for WSD. *Word Sense Disambiguation*, pp. 107–131. [88]

Miles, Alistair, Matthews, Brian, Wilson, Michael, and Brickley, Dan (2005). SKOS core: simple knowledge organisation for the web. *International Conference on Dublin Core and Metadata Applications*, pp. pp–3. [48]

Miller, George A. (1995). WordNet: a lexical database for English. *Communications of the ACM*, Vol. 38, pp. 39–41. ISSN 0001–0782. [62, 85, 86, 96, 144, 151]

Mocan, Adrian, Cimpian, Emilia, and Kerrigan, Mick (2006). Formal model for ontology mapping creation. *The Semantic Web-ISWC 2006*, pp. 459–472. Springer. [22]

Mochol, Malgorzata and Jentzsch, Anja (2008). Towards a rule-based matcher selection. *Knowledge Engineering: Practice and Patterns*, pp. 109–119. Springer. [21]

Montiel-Ponsoda, Elena, Cea, G Aguado de, Gómez-Pérez, Asunción, and Peters, Wim (2011). Enriching ontologies with multilingual information. *Natural language engineering*, Vol. 17, No. 03, pp. 283–309. [20]

Montoyo, Andres, Suárez, Armando, Rigau, German, and Palomar, Manuel (2005). Combining knowledge-and corpus-based word-sense-disambiguation methods. *Journal of Artificial Intelligence Research*, Vol. 23, No. 1, pp. 299–330. [88]

Myers, Jerome L., Well, Arnold D., and Lorch Jr., Robert F. (2010). *Research design and statistical analysis.* Routledge. [138]

Nagata, Takeshi, Watanabe, H, Ohno, M, and Sasaki, H (2000). A multi-agent approach to power system restoration. *Power System Technology, 2000. Proceedings. PowerCon 2000. International Conference on*, Vol. 3, pp. 1551–1556, IEEE. [18]

Nandi, Arnab and Bernstein, Philip A (2009). HAMSTER: using search clicklogs for schema and taxonomy matching. *Proceedings of the VLDB Endowment*, Vol. 2, No. 1, pp. 181–192. [22]

Navigli, Roberto (2009). Word sense disambiguation: A survey. *ACM Comput. Surv.*, Vol. 41, No. 2, pp. 10:1–10:69. ISSN 0360–0300. [87, 88, 90, 152]

Navigli, Roberto and Ponzetto, Simone Paolo (2010). BabelNet: Building a very large multilingual semantic network. *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pp. 216–225, Association for Computational Linguistics. [86, 151]

Nejdl, Wolfgang, Wolf, Boris, Qu, Changtao, Decker, Stefan, Sintek, Michael, Naeve, Ambjörn, Nilsson, Mikael, Palmér, Matthias, and Risch, Tore (2002). EDUTELLA: a P2P networking infrastructure based on RDF. *Proceedings of the 11th international conference on World Wide Web*, pp. 604–615, ACM. [11]

Ngo, Duy Hoa, Bellahsene, Zohra, Coletta, Remi, et al. (2011). YAM++–Results for OAEI 2011. *ISWC'11: The 6th International Workshop on Ontology Matching*, Vol. 814, pp. 228–235. [70]

Ngo, Duy Hoa, Bellahsene, Zohra, and Coletta, R. (2012). YAM++-A combination of graph matching and machine learning approach to ontology alignment task. *Journal of Web Semantic.* [67, 73, 148, 149]

Nguyen, Hung Quoc Viet, Luong, Xuan Hoai, Miklós, Zoltán, Quan, Tho Thanh, and Aberer, Karl (2013). Collaborative schema matching reconciliation. *On the Move to Meaningful Internet Systems: OTM 2013 Conferences*, pp. 222–240. Springer Berlin Heidelberg. [23]

Niles, Ian and Pease, Adam (2001). Towards a standard upper ontology. *Proceedings of the international conference on Formal Ontology in Information Systems-Volume 2001*, pp. 2–9, ACM. [21, 62, 86]

Niles, Ian and Terry, Allan (2004). The MILO: A general-purpose, mid-level ontology. *Proceedings of the International conference on information and knowledge engineering*, pp. 15–19. [62, 86]

Noy, Natalya F. (2004). Semantic integration: a survey of ontology-based approaches. *ACM Sigmod Record*, Vol. 33, No. 4, pp. 65–70. [62, 73]

Noy, Natalya F. and Musen, Mark A (2000). Algorithm and tool for automated ontology merging and alignment. *Proceedings of AAAI-00.* [116]

Noy, Natalya F. and Musen, Mark A. (2001). Anchor-PROMPT: Using non-local context for semantic matching. *Proceedings of the ICJAI workshop on ontologies and information sharing*, pp. 63–70. [63, 69, 74, 116]

Noy, Natalya F. and Musen, Mark A. (2002). Promptdiff: A fixed-point algorithm for comparing ontology versions. *AAAI/IAAI*, Vol. 2002, pp. 744–750. [10]

Noy, Natalya F. and Musen, Mark A. (2003). The PROMPT suite: interactive tools for ontology merging and mapping. *International Journal of Human-Computer Studies*, Vol. 59, No. 6, pp. 983–1024. [22, 24, 25, 74]

Noy, Natalya F. and Musen, Mark A. (2004). Ontology versioning in an ontology management framework. *Intelligent Systems, IEEE*, Vol. 19, No. 4, pp. 6–13. [10]

Noy, Natalya F., Griffith, Nicholas, and Musen, Mark A. (2008). Collecting community-based mappings in an ontology repository. *The Semantic Web-ISWC 2008*, pp. 371–386. Springer. [24]

Oard, Douglas W., Hedin, Bruce, Tomlinson, Stephen, and Baron, Jason R. (2008). Overview of the TREC 2008 legal track. Technical report, DTIC Document. [21]

Pang-Ning, Tan, Steinbach, Michael, and Kumar, Vipin (2005). *Introduction to Data Mining*. Addison Wesley, 1 edition. ISBN 0321321367. [92, 119, 155]

Parent, Christine and Spaccapietra, Stefano (1998). Issues and approaches of database integration. *Communications of the ACM*, Vol. 41, No. 5es, pp. 166–178. [3]

Paulheim, Heiko (2012). WeSeE-Match results for OEAI 2012. *Proceedings of The Seventh ISWC International Workshop on Ontology Matching(OM)*. [76]

Pedersen, Ted (2006). Unsupervised corpus-based methods for WSD. *Word Sense Disambiguation*, pp. 133–166. [88]

Pedersen, Ted, Banerjee, Satanjeev, and Patwardhan, Siddharth (2005). Maximizing semantic relatedness to perform word sense disambiguation. *University of Minnesota Supercomputing Institute Research Report UMSI*, Vol. 25, p. 2005. [90, 95]

Pipattanasomporn, Manisa, Feroze, Hassan, and Rahman, Saifur (2009). Multi-agent systems in a distributed smart grid: Design and implementation. *Power Systems Conference and Exposition, 2009. PSCE'09. IEEE/PES*, pp. 1–8, IEEE. [18]

Plessers, Peter and De Troyer, Olga (2005). Ontology change detection using a version log. *The Semantic Web–ISWC 2005*, pp. 578–592. Springer. [9]

Pouwelse, Johan, Garbacki, Paweł, Epema, Dick, and Sips, Henk (2005). The bittorrent p2p file-sharing system: Measurements and analysis. *Peer-to-Peer Systems IV*, pp. 205–216. Springer. [10]

Po, Laura and Sorrentino, Serena (2011). Automatic generation of probabilistic relationships for improving schema matching. *Information Systems*, Vol. 36, No. 2, pp. 192–208. [87, 91]

Quinlan, John Ross (1986). Induction of decision trees. *Machine learning*, Vol. 1, No. 1, pp. 81–106. [139]

Qu, Yuzhong, Hu, Wei, and Cheng, Gong (2006). Constructing virtual documents for ontology matching. *Proceedings of the 15th international conference on World Wide Web*, WWW '06, pp. 23–31, ACM, New York, NY, USA. ISBN 1–59593–323–9. [62, 75, 96, 97, 107, 108, 109, 112, 117, 141, 143, 148, 149, 150]

Raffio, Alessandro, Braga, Daniele, Ceri, Stefano, Papotti, Paolo, and Hernandez, Mauricio A (2008). Clip: a visual language for explicit schema mappings. *Data Engineering, 2008. ICDE 2008. IEEE 24th International Conference on*, pp. 30–39, IEEE. [22]

Rahm, Erhard and Bernstein, Philip A (2001). A survey of approaches to automatic schema matching. *the VLDB Journal*, Vol. 10, No. 4, pp. 334–350. [57, 58, 63, 73, 116]

Rahm, Erhard, Do, HongHai, and Massmann, Sabine (2004). Matching large XML schemas. *ACM SIGMOD Record*, Vol. 33, No. 4, pp. 26–31. [64]

Redmond, Timothy, Smith, Michael, Drummond, Nick, and Tudorache, Tania (2008). Managing Change: An Ontology Version Control System. *OWLED*. [9]

Resnik, Philip and Yarowsky, David (1999). Distinguishing systems and distinguishing senses: New evaluation methods for word sense disambiguation. *Natural language engineering*, Vol. 5, No. 02, pp. 113–133. [90]

Rijsbergen, Cornelis J. Van (1979). Information Retrieval. ISBN 0408709294. [36, 38]

Ritze, Dominique and Eckert, Kai (2012). Thesaurus mapping: a challenge for ontology alignment? *Proceedings of The Seventh International Workshop on Ontology Matching (OM-2012) collocated with the 11th International Semantic Web Conference (ISWC-2012)*, pp. 248–249. [48]

Rogozan, Delia and Paquette, Gilbert (2005). Managing ontology changes on the semantic web. *Web Intelligence, 2005. Proceedings. The 2005 IEEE/WIC/ACM International Conference on*, pp. 430–433, IEEE. [9]

Rosse, Cornelius and Mejino Jr, José LV (2003). A reference ontology for biomedical informatics: the Foundational Model of Anatomy. *Journal of biomedical informatics*, Vol. 36, No. 6, pp. 478–500. [62]

Rumelhart, David E., Hinton, Geoffrey E., and Williams, Ronald J. (1986). Learning representations by back-propagating errors. *Nature*, Vol. 323, pp. 533–536. [67]

Russell, Stuart J. and Norvig, Peter (2003). *Artificial Intelligence: A Modern Approach*. Pearson Education, 2 edition. ISBN 0137903952. [66, 67]

Sabou, Marta, d'Aquin, Mathieu, and Motta, Enrico (2008). Exploring the Semantic Web as Background Knowledge for Ontology Matching. *Journal on Data Semantics XI*, Vol. 5383 of *Lecture Notes in Computer Science*, pp. 156–190. Springer Berlin Heidelberg. ISBN 978–3–540–92147–9. [21, 63, 116]

Salton, Gerard and Buckley, Christopher (1988). Term-weighting approaches in automatic text retrieval. *Information Processing and Management*, Vol. 24, No. 5, pp. 513 – 523. ISSN 0306–4573. [100]

Salton, Gerard, Wong, Anita, and Yang, Chung-Shu (1975). A vector space model for automatic indexing. *Communications of the ACM*, Vol. 18, pp. 613–620. ISSN 0001–0782. [92]

Saruladha, K., Aghila, G., and Sathiya, B. (2011). A Comparative Analysis of Ontology and Schema Matching Systems. *International Journal of Computer Applications*, Vol. 34, No. 8, pp. 14–21. Published by Foundation of Computer Science, New York, USA. [86]

Schadd, Frederik C. and Roos, N. (2011a). Improving ontology matchers utilizing linguistic ontologies: an information retrieval approach. *Proceedings of the 23rd Belgian-Dutch Conference on Artificial Intelligence (BNAIC 2011)*, pp. 191–198. [83]

Schadd, Frederik C. and Roos, N. (2011b). Maasmatch results for OAEI 2011. *Proceedings of The Sixth International Workshop on Ontology Matching (OM-2011) collocated with the 10th International Semantic Web Conference (ISWC-2011)*, pp. 171–178. [83]

Schadd, Frederik C. and Roos, N. (2012a). Coupling of WordNet Entries for Ontology Mapping using Virtual Documents. *Proceedings of The Seventh International Workshop on Ontology Matching (OM-2012) collocated with the 11th International Semantic Web Conference (ISWC-2012)*, pp. 25–36. [83]

Schadd, Frederik C. and Roos, N. (2012b). Maasmatch results for OAEI 2012. *Proceedings of The Seventh ISWC International Workshop on Ontology Matching*, pp. 160–167. [119]

Schadd, Frederik C. and Roos, N. (2013). Anchor-Profiles for Ontology Mapping with Partial Alignments. *Proceedings of the 12th Scandinavian AI Conference (SCAI 2013)*, pp. 235–244. [45, 115]

Schadd, Frederik C. and Roos, N. (2014a). Anchor-Profiles: Exploiting Profiles of Anchor Similarities for Ontology Mapping. *Proceedings of the 26th Belgian-Dutch Conference on Artificial Intelligence (BNAIC 2014)*, pp. 177–178. [115]

Schadd, Frederik C. and Roos, Nico (2014b). Word-Sense Disambiguation for Ontology Mapping: Concept Disambiguation using Virtual Documents and Information Retrieval Techniques. *Journal on Data Semantics*, pp. 1–20. ISSN 1861–2032. http://dx.doi.org/10.1007/s13740-014-0045-5. [83, 141]

Schadd, Frederik C. and Roos, Nico (2014c). A Feature Selection Approach for Anchor Evaluation in Ontology Mapping. *Knowledge Engineering and the Semantic Web*, pp. 160–174. Springer International Publishing. [133]

Schadd, Frederik C. and Roos, N. (2015). Matching Terminological Heterogeneous Ontologies by Exploiting Partial Alignments. *Proceedings of the 9th International Conference on Advances in Semantic Processing (SEMAPRO 2015)*, pp. 13–19. **Best Paper Award.** [147]

Schütze, Hinrich (1992). Dimensions of meaning. *Proceedings of the 1992 ACM/IEEE Conference on Supercomputing*, **pp. 787–796, IEEE.** [90]

Schütze, Hinrich and Pedersen, Jan O (1995). Information Retrieval Based on Word Senses. *In Proceedings of the 4th Annual Symposium on Document Analysis and Information Retrieval*. [88]

Seddiqui, Md Hanif and Aono, Masaki (2009). An efficient and scalable algorithm for segmented alignment of ontologies of arbitrary size. *Web Semantics: Science, Services and Agents on the World Wide Web*, **Vol. 7, No. 4, pp. 344–356.** [63, 75, 84, 116, 135]

Sheth, Amit P. and Larson, James A. (1990). Federated database systems for managing distributed, heterogeneous, and autonomous databases. *ACM Computing Surveys (CSUR)*, **Vol. 22, No. 3, pp. 183–236.** [3]

Sheth, Amit P., Gala, Sunit K., and Navathe, Shamkant B. (1993). On automatic reasoning for schema integration. *International Journal of Intelligent and Cooperative Information Systems*, **Vol. 2, No. 01, pp. 23–50.** [3]

Shokouhi, Milad and Si, Luo (2011). Federated search. *Foundations and Trends in Information Retrieval*, **Vol. 5, No. 1, pp. 1–102.** [7]

Shvaiko, Pavel and Euzenat, Jérôme (2005). A Survey of Schema-Based Matching Approaches. *Journal on Data Semantics IV*, **Vol. 3730, pp. 146–171. Springer. ISBN 978–3–540–31001–3.** [57, 58, 86, 119, 134]

Shvaiko, Pavel and Euzenat, Jérôme (2008). Ten Challenges for Ontology Matching. *Proceedings of ODBASE 2008*, **pp. 1164—-1182.** [19, 47]

Shvaiko, Pavel and Euzenat, Jérôme (2013). Ontology Matching: State of the Art and Future Challenges. *Knowledge and Data Engineering, IEEE Transactions on*, **Vol. 25, No. 1, pp. 158 –176. ISSN 1041–4347.** [19, 47]

Shvaiko, Pavel, Giunchiglia, Fausto, Da Silva, Paulo Pinheiro, and McGuinness, Deborah L. (2005). Web explanations for semantic heterogeneity discovery. *The Semantic Web: Research and Applications*, **pp. 303–317. Springer.** [22]

Sicilia, Miguel A., Garcia, Elena, Sanchez, Salvador, and Rodriguez, Elena (2004). On integrating learning object metadata inside the OpenCyc knowledge base. *Advanced Learning Technologies, 2004. Proceedings. IEEE International Conference on*, **pp. 900–901, IEEE.** [86]

Solimando, Alessandro, Jiménez-Ruiz, Ernesto, and Pinkel, Christoph (2014). Evaluating Ontology Alignment Systems in Query Answering Tasks. *International Semantic Web Conference (ISWC). Poster track.* **[50]**

Spaccapietra, Stefano and Parent, Christine (1994). View integration: A step forward in solving structural conflicts. *Knowledge and Data Engineering, IEEE Transactions on*, **Vol. 6, No. 2, pp. 258–274. [3]**

Sparck Jones, Karen (1972). A statistical interpretation of term specificity and its application in retrieval. *Journal of documentation*, **Vol. 28, No. 1, pp. 11–21. [100]**

Sproat, Richard, Hirschberg, Julia, and Yarowsky, David (1992). A corpus-based synthesizer. *Proceedings of the International Conference on Spoken Language Processing*, **Vol. 92, pp. 563–566. [89]**

Stoilos, Giorgos, Stamou, Giorgos, and Kollias, Stefanos (2005). A string metric for ontology alignment. *The Semantic Web–ISWC 2005*, **pp. 624–637. Springer. [75]**

Strube, Michael and Ponzetto, Simone Paolo (2006). WikiRelate! Computing semantic relatedness using Wikipedia. *AAAI*, **Vol. 6, pp. 1419–1424. [62, 85]**

Stuckenschmidt, Heiner and Klein, Michel (2004). Structure-based partitioning of large concept hierarchies. *The Semantic Web–ISWC 2004*, **pp. 289–303. [53]**

Subrahmanian, V.S., Adali, Sibel, Brink, Anne, Emery, Ross, Lu, James J, Rajput, Adil, Rogers, Timothy J, Ross, Robert, and Ward, Charles (1995). HERMES: A heterogeneous reasoning and mediator system. [5, 8]

Suchanek, Fabian M, Kasneci, Gjergji, and Weikum, Gerhard (2007). Yago: a core of semantic knowledge. *Proceedings of the 16th international conference on World Wide Web*, **pp. 697–706, ACM. [62]**

Suchanek, Fabian M., Kasneci, Gjergji, and Weikum, Gerhard (2008). Yago: A large ontology from wikipedia and wordnet. *Web Semantics: Science, Services and Agents on the World Wide Web*, **Vol. 6, No. 3, pp. 203–217. [85]**

Sure, York, Staab, Steffen, and Studer, Rudi (2002). Methodology for development and employment of ontology based knowledge management applications. *ACM SIGMOD Record*, **Vol. 31, No. 4, pp. 18–23. [153]**

Su, Xiaomeng and Gulla, Jon Atle (2004). Semantic Enrichment for Ontology Mapping. *Natural Language Processing and Information Systems: 9th International Conference on Applications of Natural Languages to Information*

Systems, NLDB 2004, Salford, UK, June 23-25, 2004, Proceedings, Vol. 3136, p. 217, Springer. [149]

Sycara, Katia, Paolucci, Massimo, Ankolekar, Anupriya, and Srinivasan, Naveen (2003). Automated discovery, interaction and composition of semantic web services. *Web Semantics: Science, Services and Agents on the World Wide Web*, Vol. 1, No. 1, pp. 27–46. [13]

Thornton, Chris (1998). Separability is a learner's best friend. *4th Neural Computation and Psychology Workshop, London, 9–11 April 1997*, pp. 40–46, Springer. [139]

Tran, Thanh, Cimiano, Philipp, Rudolph, Sebastian, and Studer, Rudi (2007). Ontology-based interpretation of keywords for semantic search. *The Semantic Web*, pp. 523–536. Springer. [16]

Trojahn, Cássia, Meilicke, Christian, Euzenat, Jérôme, and Stuckenschmidt, Heiner (2010). Automating oaei campaigns (first report). *Proceedings of the International Workshop on Evaluation of Semantic Technologies (IWEST).* [46]

Trojahn, Cássia, Euzenat, Jérôme, Tamma, Valentina, and Payne, Terry R. (2011). Argumentation for reconciling agent ontologies. *Semantic Agent Systems*, pp. 89–111. Springer. [19, 23]

Valtchev, Petko and Euzenat, Jérôme (1997). Dissimilarity measure for collections of objects and values. *Advances in Intelligent Data Analysis Reasoning about Data*, pp. 259–272. Springer. [61]

Wache, Holger, Voegele, Thomas, Visser, Ubbo, Stuckenschmidt, Heiner, Schuster, Gerhard, Neumann, Holger, and Hübner, Sebastian (2001). Ontology-based integration of information-a survey of existing approaches. *IJCAI-01 workshop: ontologies and information sharing*, Vol. 2001, pp. 108–117. [73]

Walker, Jan, Pan, Eric, Johnston, Douglas, Adler-Milstein, Julia, Bates, David W, and Middleton, Blackford (2005). The value of health care information exchange and interoperability. *HEALTH AFFAIRS-MILLWOOD VA THEN BETHESDA MA-*, Vol. 24, p. W5. [204, 205]

Wang, Jun and Gasser, Les (2002). Mutual online ontology alignment. *Proceedings of the Workshop on Ontologies in Agent Systems, held with AAMAS 2002.* [19]

Wang, Shenghui, Englebienne, Gwenn, and Schlobach, Stefan (2008). Learning concept mappings from instance similarity. *The Semantic Web-ISWC 2008*, pp. 339–355. Springer. [14, 60]

Wang, Chang, Kalyanpur, Aditya, Fan, James, Boguraev, Branimir K., and Gondek, DC (2012). Relation extraction and scoring in DeepQA. *IBM Journal of Research and Development*, Vol. 56, No. 3.4, pp. 9–1. [17]

Watters, Carolyn (1999). Information retrieval and the virtual document. *Journal of the American Society for Information Science*, **Vol. 50, pp. 1028–1029. ISSN 0002–8231.** [89]

Weaver, Warren (1955). Translation. *Machine translation of languages*, **Vol. 14, pp. 15–23.** [88]

Wiesman, Floris and Roos, Nico (2004). Domain independent learning of ontology mappings. *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems-Volume 2*, **pp. 846–853, IEEE Computer Society.** [19]

Wiesman, Floris, Roos, Nico, and Vogt, Paul (2002). Automatic ontology mapping for agent communication. *Proceedings of the first international joint conference on Autonomous agents and multiagent systems: part 2*, **pp. 563–564, ACM.** [19]

Wilkes, Yorick (1975). Preference semantics. *Formal Semantics of Natural Language* **(ed. E.L. Keenan), pp. 329–348., Cambridge University Press.** [88]

Xiao, Bo and Benbasat, Izak (2007). E-commerce product recommendation agents: use, characteristics, and impact. *Mis Quarterly*, **Vol. 31, No. 1, pp. 137–209.** [18]

Yarowsky, David (1994). Decision lists for lexical ambiguity resolution: Application to accent restoration in Spanish and French. *Proceedings of the 32nd annual meeting on Association for Computational Linguistics*, **pp. 88–95, Association for Computational Linguistics.** [89]

Zhdanova, Anna V. and Shvaiko, Pavel (2006). Community-driven ontology matching. *The Semantic Web: Research and Applications*, **pp. 34–49. Springer.** [23]

List of Figures

List of Tables

List of Algorithms

Addendum: Valorization

Remark: This addendum is required by the regulation governing the attainment of doctoral degrees (2013) of Maastricht University. As stated there, the addendum "does not form part of the dissertation and should not be assessed as part of the dissertation".

The ability to transfer data between information systems, also referred to as *interoperability* between systems, presents an ever-growing issue in a society which adopts electronic solutions in a growing range of domains. If this data is not standardized, it becomes necessary to apply a transformation to the data, based on a given mapping, in order to make the data transfer possible. In section 1.2 we introduced a wide range of applications for the presented research of this thesis, where it is required for data to be exchanged between information systems. These application include schema integration, information integration, ontology engineering, information sharing, web-service composition, querying of semantic information and agent communication.

In the following sections, we will introduce three real-world domains, namely (1) the US Capital Facility Industry, (2) the US Automotive Industry and (3) the US Health Care Industry, which regularly face interoperability issues. These issues are typically resolved via conventional means, resulting in operational inefficiencies and added costs. Examples of such conventional means are transforming and entering data into different systems by hand, redesigning information systems due to incompatibility or outsourcing information exchange responsibilities to third parties. We present the results of three scientific studies which attempted to quantify the annual costs, that these domains are forced to compensate for, due to unresolved interoperability issues. Interoperability costs are compiled by estimating factors such as added labour costs of data transformation and verification, added labour costs of reworking and redesigning ongoing projects due to unexpected incompatibilities, purchase costs of new systems and resources, delay costs and lost revenue costs.

Cost Estimate: US Capital Facility Industry

The so called *Capital Facility* industry is a component of the entire US construction industry. The core activities of this industry encompass the design, construction and maintenance of large buildings, facilities and plants. These buildings are ordered by commercial, industrial and institutional sectors. Due to the large scale of typically requested buildings, the capital facility industry has large data requirements. Examples of data exchange in this sector is the sharing of data among all stakeholders, possibly across several information systems, and the integration of multi-vendor equipment and systems. Due to these requirements, the capital facility industry is particularly vulnerable to interoperability issues.

In 2004, the Building and Fire Research Laboratory and the Advanced Technology Program at the National Institute of Standards and Technology (NIST) issued a study to estimate the inefficiencies caused by interoperability issues between computer-aided design, engineering and software systems. This study has been performed by RTI International and the Logistic Management Institute (Gallaher *et al.*, 2004). The following stakeholders were identified which typically face direct interoperability issues during the execution of a project:

Architects and Engineers covering architects, general and speciality engineers, and facilities consultancies.

General Contractors covering general contractors tasked with physical construction and project management.

Speciality Fabricators and Suppliers covering speciality constructors and systems suppliers, including elevators, steel, and HVAC systems.

Owners and Operators covering the entities that own and/or operate the facilities.

Participants from each stakeholder group contributed to the study through interviews with the experimenters or by completing surveys. The participants were tasked to quantify their incurred interoperability costs by listing which activities they perform in order to resolve these issues. By extrapolating the costs associated with these activities, a cost estimate could then be established. The activities and their associated costs were grouped into three categories: (1) avoidance costs, (2) mitigation costs and (3) delay costs. Examples of avoidance costs are the costs of outsourcing of translation services to third parties, investing in in-house programs and the costs of purchasing, maintaining and training for redundant computer-aided design and engineering systems. Mitigation costs typically involve the costs associated with rework of designs or construction, re-entering data when automated transfer systems fail and information verification. Examples of delay costs are costs of idle resources due to delays, lost profits due to delay in revenues and losses to customers and consumers due to project delays. An overview of the estimated costs for each stakeholder group can be viewed in Table A.

Stakeholder Group	Avoidance Costs	Mitigation Costs	Delay Costs	Total
Architects and Engineers	485,3	684.5	-	1,169.8
General Contractors	1,095.4	693.3	13.0	1,801.7
Speciality Fabricators and Suppliers	1,908.4	296.1	-	2,204.5
Owners and Operators	3,120.0	6,028.2	1,499.8	10,648.0
All Stakeholders	6,609.1	7,702.0	1,512.8	15,824.0

Table A: Annual cost of inadequate interoperability in the US capital facility industry by cost category, by stakeholder group (in $Millions)(Gallaher *et at.,* 2004).

As we can see in Table A, the capital facility industry has to compensate for substantial interoperability costs. The total costs are estimated at $15.8 billion annually, which corresponds to approximately 3-4% of the entire industry's annual revenue.

Cost Estimate: US Automotive Industry

In 2002, the Research Triangle Institute conducted a study for the National Institute of Standards and Technology (NIST) in order to quantify to what degree the US Automotive supply chain suffers from interoperability issues (Brunnermeier and Martin, 2002). Similar to the previous study, the experimenters surveyed different stakeholders across the industry about typically faced interoperability problems. The costs of the provided answers are extrapolated in order to estimate the severity of these costs across the entire automotive industry. This estimate is referred to as the *Cost Component Approach*. The experimenters also interviewed several key industry executives about their viewpoints. The executives provided their own estimates of the incurred interoperability costs, allowing for the inclusion of costs which might not have been considered by the experimenters. This method of cost estimation of referred to as the *Aggregate Cost Approach*. An additional benefit of consulting industry executives is that it validates the results of the *Cost Component Approach* if their results are similar to a certain degree. The cost estimates of both approaches can be viewed in Table B.

The results of both estimates depict that the automotive industry suffers significant monetary losses due to interoperability issues. According to the *Cost Component Approach* $1.05 billion are lost yearly, while the *Aggregate Cost Approach* resulted in a cost estimate of $1.015 billion.

Source of Cost	Annual Cost ($Millions)	Percentage
Cost Component Approach		
Avoidance Costs	52.8	5
Mitigation Costs	907.6	86
Delay Costs	90.0	9
Total	1,050.4	100
Aggregate Cost Approach		
Interoperability Cost	925.6	91
Delayed Profits	90.0	9
Total	1,015.6	100

Table B: Aggregate of estimated annual interoperability costs of the US automotive industry (Brunnermeier and Martin, 2002).

Net Value Estimate: US Health Care Industry

The health care sector sees an every increasing use of information technology. Commonly used technologies in this sector are the storage of patient data in electronic medical records, computerized physician order entry systems, and decision support tools. Facilitating easy information exchange between these systems would result in lower transaction costs of information exchange, increased operating efficiency and a higher quality of service due to fewer transaction mistakes and easier access to critical medical data. Additionally, most healthcare facilities still store patient information on paper-based formats. Therefore, every time paper-based data needs to be transferred to a different stakeholder, it is necessary that it is entered into the information system by hand, resulting in a huge operating inefficiency.

Walker *et al.* (2005) investigated what the net value of a fully implemented health care information exchange and interoperability (HIEI) system would be. This study weighed the estimated interoperability cost savings of the US health care domain against the estimated project costs of a full roll-out of a HIEI system. The study defined four levels of interoperability and estimated the net value of achieving each level. The levels are defined as follows:

Level 1: Non-electronic data. No use of IT to share information. This level represents the operational efficiency of the health care system prior to the introduction of IT and serves as *baseline* comparison for determining the benefits of the other levels.

Level 2: Machine transportable data. Transfer on non-standardized data via basic IT channels. Data cannot be manipulated by machines (e.g. exchange of scanned documents via fax or PDF files).

Level 3: Machine organizable data. Transfer of non-standardized data via structured messages. Requires mappings such that data conforming to different standards can be interpreted by each local system. Still requires transferred data to be verified due to the risk of imperfect mappings.

Level 4: Machine interpretable data. Transfer of standardized data via structured messages, allowing data to be transferred to and understood by all local systems.

Level 2 systems are already universally implemented amongst all health-care institutions, therefore requiring no costs of implementation. The costs of adopting level 3 and 4 were estimated by compiling various cost estimates for the sub-components of each level from different sources, such as established scientific studies, the US census bureau and expert-panel judgements (Walker *et al.*, 2005). The aggregate cost estimates of the HIEI implementations and their resulting net values are listed in Table C.

	Implementation, cumulative years 1–10	Steady state, annual starting year 11
Level 2		
Benefit	141.0	21.6
Cost	0.0	0.0
Net Value	141.0	21.6
Level 3		
Benefit	286	44.0
Cost	320	20.2
Net Value	–34.2	23.9
Level 4		
Benefit	613	94.3
Cost	276	16.5
Net Value	337	77.8

Table C: Estimated net value of deployment of HIEI systems, according to different levels of sophistication, in the US health care industry(in $Billions)(Walker *et al., 2005*).

The result of the study indicated that a nationwide adoption of level 4 HIEI system would result in an annual net gain of $77.8 billion after the initial implementation phase. This corresponds to approximately 5% of the annual US expenditures on health care.

Summary

The availability of data plays an ever increasing role in our society. Businesses commonly store information about their customers, transactions and products in large information systems. This allows them to analyse their data to gain more knowledge, such as trends and predictions, in order to improve their business strategy. Furthermore, the core strategy of a business can be built on enabling the user to easily access a certain type of data. Such services play an increasing role in common every-day life. For example, services such as *Google* and *Wikipedia* are widely used to find general information, whereas services such as *Amazon*, *bol.com* and *Yelp* are used to find information and reviews about products. Some of these site also allow the user to purchase the queried products on the same site. To be able to interpret stored data, it is necessary that the data is structured and annotated with meta-information, such that for each data entry it is possible to determine its meaning and relation to other data entries. For example, a data entry '555-12345' has very little use if it is not known that it represents a telephone number and who the owner of the number is. An information system specifies this type of meanings and their structure using an ontology. An ontology specifies the types of objects, referred to as concepts, about which one intends to store information, what kind of data is stored for each concept and how the concepts are related to each other.

A common problem faced by businesses is the desire to be able to exchange information between different systems. An example scenario would be *Company A* deciding to acquire *Company B*. To continue the operations of *Company B*, *Company A* would need to transfer all the data of the information system of *Company B* into its own information system. Here, it can occur that the data in the information systems of both companies is modelled using different ontologies. This can stem from the companies having different requirements for their systems or having followed separate design principles in the creation of their ontologies. In this case, it is not possible to simply transfer data between the systems since these are incompatible.

A possible solution for enabling the exchange of information between systems utilizing different ontologies is the process of *ontology mapping*. Ontology mapping aims to identify all pairs of concepts between two ontologies which are used to model the same type of information. A full list of correspondences between two ontologies is known as an alignment or mapping. Based on such a mapping, it is possible to create a transfer function such that every data entry part of one ontology can be re-formulated such that it conforms to the specification of the other ontology. This allows for the transfer of data between two information systems despite the systems

using different ontology structures.

Mapping ontologies is a labour intensive task. To create a mapping, a domain expert has to manually define and verify every correspondence. This approach is infeasible when having to map large ontologies encompassing thousands of concepts. Hence, automatic approaches to ontology mapping are required in order to solver interoperability problems in the corporate domain. A different domain of application is the *Semantic Web*. This domain envisions the next step in the evolution of the world-wide-web, where all available information is machine readable and semantically structured. This semantic structure is also specified using an ontology and allows machines to gather semantic information from the web. However, in order to retrieve semantic information autonomously, a machine needs to be capable to also autonomously match ontologies. This is necessary such that the machine can query sources which represent their information using a different semantic structure.

Ontology mapping has been an active field of research in the past decade. Here, matching systems typically utilize a combination of techniques to determine the similarity between concepts. From these computations, highly similar concepts are extracted which then form the alignment between the given ontologies. In some situations, it is possible that an extra resource of information is available that can be exploited to aid the matching process. An example of such extra information are lexical resources, for instance *Wikipedia*. A lexical resource allows a system to look up word definitions, identify synonyms and look up information of related concepts. A different example resource are partial alignments. A partial alignment is an incomplete mapping stemming from an earlier matching effort. It can be the result of a domain expert attempting to create a mapping, but being unable to finish it due to time constraints. A core challenge within the field of ontology mapping thus is to devise techniques which can use these resources for the purpose of creating a complete mapping. This has led us to the following problem statement:

> *How can we improve ontology mapping systems by exploiting auxiliary information?*

To tackle this problem statement, we formulated four research questions upon which we based our research:

1. *How can lexical sense definitions be accurately linked to ontology concepts?*

2. *How can we exploit partial alignments in order to derive concept correspondences?*

3. *How can we evaluate whether partial alignment correspondences are reliable?*

4. *To what extent can partial alignments be used in order to bridge a large terminological gap between ontologies?*

In Chapter 1 we introduce the reader to the field of ontology mapping. Here, we introduce the problems that arise when attempting to transfer data between knowledge systems with different ontologies. Further, we present a series of real-world domains which can benefit from the research of this thesis, such as information integration, web-service composition and agent communication. We also present a brief overview of the core research challenges of the field of ontology mapping. In the final section of the chapter, we introduce and discuss the problem statement and research questions which guide this thesis.

Chapter 2 provides important background information to the reader. We formally introduce the problem of ontology matching. Further, we detail and illustrate common techniques that are applied for the purpose of ontology alignment evaluation. Lastly, we introduce a series of datasets which can be used in order to evaluate ontology matching systems.

Applicable techniques to ontology matching are introduced in Chapter 3. Here, we introduce the reader to contemporary ontology matching systems and their underlying architectures. We introduce the three core tasks that a matching systems has to perform, being similarity computation, similarity combination and correspondence extraction, and provide an overview of techniques which are applicable for these respective tasks. Additionally, we provide a brief survey of existing ontology matching systems with the focus on systems utilizing auxiliary resources.

Chapter 4 answers the first research question. Here, the core problem concerns the linking of correct lexical definitions to the modelled concepts of an ontology, referred to as concept disambiguation. An example of such an action is determining that the concept name 'Plane' refers to the type of mathematical surfaces instead of the type of airborne vehicles. Techniques utilizing lexical resources rely on these links to determine concept similarities using various techniques. We tackle this research question by proposing an information retrieval based method for associating ontology concepts with lexical senses. Using this method, we define a framework for the filtering of concept senses based on sense-similarity scores and a given filtering policy. We evaluate four filtering policies which filter senses if their similarity scores resulted in an unsatisfactory value. The filtering policies are evaluated using several lexical similarities in order to investigate the general effects of the filtering policies on the lexical similarities. Our evaluation revealed that the application of our disambiguation approach improved the performances of all lexical metrics. Additionally, we investigated the effect of weighting the terms of the sense annotations and concept annotations. This evaluation revealed that weighting terms according to respective origins within the ontologies or lexical resource resulted in a superior performance compared to weighting the document terms using the widely used TF-IDF approach from the field of information retrieval.

The research question tackled in Chapter 5 concerns the exploitation of partial alignments. The core problem here is that a matching system is given an incomplete alignment, referred to as partial alignment, and has to compute the remaining correspondences to create a full mapping. For this purpose, one has to create mapping techniques which utilize the information of the individual partial alignment correspondences, referred to as *anchors*, in order to improve the mapping quality. To answer this question, we propose a method which compares concepts by measuring their similarities with the provided anchor concepts. For each concept, its measurements are compiled into an *anchor-profile*. Two concepts are then considered similar if their anchor-profiles are similar, i.e. they exhibit comparable degrees of similarity towards the anchor concepts. The evaluation revealed that the application of our approach can result into performances similar to top matching systems in the field. However, we observe that the performance depends on the existence of appropriate meta-information which is used to compare concepts with anchors. From this, we conclude that a combination of similarity metrics, such that all types of meta-information are exploited, should be used to ensure a high performance for all types of matching problems. Lastly, we systematically investigate the effect of the partial alignment size and correctness on the quality of the produced alignments. We observe that both size and correctness have a positive influence on the alignment quality. We observe that decreasing the degree of correctness has a more significant impact on the alignment quality than decreasing the size. From this we conclude that matching systems exploiting partial alignments need to take measures to ensure the correctness of a given partial alignment from an unknown source.

Chapter 6 addresses research question 3, which presents the problem of ensuring the correctness of a partial alignment. Some techniques exploit partial alignments for the purpose of ontology mapping. An example of such a technique is the approach presented in Chapter 5. In order for these techniques to function correctly, it is necessary that the given partial alignment contains as few errors as possible. To evaluate the correctness of a given partial alignment, we propose a method utilizing feature evaluation techniques from the field of machine learning. To apply such techniques, one must first define a feature space. A feature space is a core mathematical concept describing a space that is spanned by inspecting n different variables. For example, taking the variables 'height', 'width' and 'depth' would span a 3-dimensional feature space. Plotting the respective values for each feature of different objects would thus allow us to inspect the differences between the objects with regard of their physical location and perform analytical tasks based on this data. In the field of machine learning, a feature space is not restricted by the amount or types of features. Therefore, one can span a feature space using any amount of features modelling quantities such as position, size, age, cost, type or duration. A core task in the field of machine learning is classification, where one must designate a class to an object for which the values for each feature are known. An example of such a task is determining whether a person is a reliable debtor given his or her income, employment type, age and marital status. A classification system does this by first analysing a series of objects for which the class values are already known. Feature evaluation techniques help the designer of a specific classification system to determine the quality of a feature with respect to the classification task by analysing the pre-classified objects. For our approach, we utilize this work-flow in order to design an evaluation system for a series of anchors. We span a feature space where every feature represents the result of a consistency evaluation between a specific anchor and a given correspondence. Using a selection of feature evaluation techniques, we then measure the quality of each feature and therefore the quality of its corresponding anchor. To generate the consistency measurements, we define a metric requiring a base similarity, for which we evaluate three types of similarities: a syntactical, a profile and a lexical similarity. For each type of similarity, we evaluate our approach against a baseline ranking, which is created by directly applying the same similarity on each anchor. Our evaluation revealed that our approach was able to produce better anchor evaluations for each type of similarity metric than the corresponding baseline. For the syntactic and lexical similarities we observed significant improvements.

The research presented in Chapter 7 tackles the fourth research question. This chapter focuses on a specific kind of matching problems, being ontologies which have very little terminology in common. Many matching techniques rely on the presence of shared or very similar terminology in order to decide whether two concepts should be matched. These techniques fail to perform adequately if the given ontologies use different terminology to model the same concepts. Existing techniques circumvent this problem by adding new terminology to the concept definitions. The new terms can be acquired by searching a lexical resource such as *WordNet*, *Wikipedia* or *Google*. However, if an appropriate source of new terminology is not available then it becomes significantly harder to match these ontologies. We investigate a possible alternative by proposing a method exploiting a given partial alignment. Our approach is built upon an existing profile similarity. This type of similarity exploits semantic relations in order to gather context information which is useful for matching. Our extension allows it to exploit the semantic relations that are specified in the partial alignment as well. The evaluation reveals that our approach can compute correspondences of similar quality exploiting only partial alignments as existing frameworks using appropriate lexical resources. Furthermore, we establish that a higher performance is achievable if both lexical resources and partial alignments are exploited by a mapping system.

Chapter 8 provides the conclusions of this thesis and discusses possibilities of future research. Taking the answers to the research questions into account, we conclude that there are a multitude of ways in which auxiliary resources can be exploited in order to aide ontology matching systems. First, lexical resources can be effectively exploited when applying a virtual document-based disambiguation policy. Second, through the creation of anchor-profiles it is possible to exploit partial alignments to derive similarity scores between concepts. Third, by using a feature-evaluation approach one can evaluate anchors to ensure approaches utilizing partial alignments perform as expected. Fourth, by extending profile similarities, such that these also exploit anchors, one can match ontologies with little to no terminological overlap.

When discussing future research, we identify several key areas which should be investigated to improve the applicability of the presented work. First, research efforts should be directed into the robustness of the approaches. For example, the disambiguation approach of Chapter 4 relies on the presence of terminological information to be able to identify senses. If this information is sparse or lacking all together, then the effectiveness of this approach can be affected. A solution could be the combination of multiple disambiguation approaches. Based on the available meta-information, a decision system could determine which approach is best suited for each ontology. A different area for future research would be the generation of reliable partial alignments. If a partial alignment does not exist for a given matching problem, then generating one during run-time would enable a matching system to use techniques which require the existence of such alignments. This would allow for the presented techniques of this thesis to be applicable to a wider group matching problems.

Samenvatting

De beschikbaarheid van data speelt een steeds belangrijkere rol in onze samenleving. Bedrijven gebruiken vaak informatiesystemen om informatie op te slaan over hun klanten, transacties en producten. Daardoor is het mogelijk om deze data te analyseren en om zo meer kennis te (her)gebruiken door bijvoorbeeld voorspellingen te doen aan de hand van trends, zodanig dat bedrijven hun handelsstrategieën kunnen verbeteren. Sterker nog, een bedrijf kan zich richten op het toegankelijk maken van databronnen voor consumenten. Dit soort diensten speelt een steeds grotere rol in het alledaagse leven. Diensten zoals *Google* en *Wikipedia* worden doorgaans gebruikt om algemene informatie te vinden. Gespecialiseerde diensten zoals *Amazon*, *bol.com* en *Yelp* worden gebruikt om informatie en beoordelingen van producten te vinden en om deze producten zelfs te kopen. Om opgeslagen data te kunnen interpreteren is het noodzakelijk dat deze data een structuur heeft en geannoteerd is met meta-informatie. Dit maakt het mogelijk om de betekenis van ieder datapunt, en zijn relatie met andere datapunten, te bepalen. Bijvoorbeeld, het datapunt '555-12345' is van weinig nut als het niet bekend is dat dit een telefoonnummer representeert en wie de eigenaar van dit nummer is. Een informatiesysteem beschrijft de betekenissen en structuur van de opgeslagen data met behulp van een zogenaamde ontologie. Deze ontologie specificeert een aantal typen, ookwel concepten genoemd, en hoe deze concepten aan elkaar zijn gerelateerd.

Bedrijven staan vaak voor het probleem dat ze informatie willen uitwisselen tussen verschillende systemen. Veronderstel bijvoorbeeld dat *Bedrijf A* beslist om *Bedrijf B* over te nemen. Om de bedrijfsvoering van *Bedrijf B* voort te kunnen zetten moet *Bedrijf A* alle informatie uit het systeem van *Bedrijf B* in zijn eigen systeem overzetten. Hier kan het gebeuren dat de data van *Bedrijf B* met een andere ontologie is gemodelleerd dan de data van *Bedrijf A*. De oorzaak hiervan kan zijn dat de twee bedrijven verschillende eisen hebben voor hun systemen of verschillende ontwerpprincipes hebben gehanteerd bij het definiëren van de ontologieën. In dit soort gevallen is het door de incompatibiliteit van de systemen niet zomaar mogelijk om data tussen de twee systemen uit te wisselen.

Een mogelijke oplossing om het overzetten van data tussen twee systemen, welke verschillende ontologieën gebruiken, mogelijk te maken is het zogenoemde *ontologie-mapping* proces. Het doel van ontologie-mapping is het identificeren van alle conceptparen welke gebruikt kunnen worden om dezelfde soort data te modelleren. Een volledige lijst van correspondenties tussen twee ontologieën wordt opgeslagen in een zogenaamde *alignment* of *mapping*. Met behulp van deze mapping is het mogelijk om data uit één ontologie te herschrijven zodat deze conform is aan de specificaties van de andere ontologie. Hierdoor wordt het mogelijk om data tussen twee systemen uit te wisselen, ondanks het feit dat de twee systemen verschillende ontologieën gebruiken. Het maken van zo'n mapping vergt veel werk. Om een mapping te maken moet een domeinexpert handmatig alle correspondenties definiëren en controleren. Deze aanpak is niet haalbaar als men een mapping tussen twee grote ontologieën moet maken, waarbij iedere ontologie duizenden concepten modelleert. Het is dus nodig om het proces van ontologie-mapping te automatiseren. Een ander applicatiedomein is het zogenaamde *Semantic Web*. Dit domein stelt de volgende stap in de evolutie van het world-wide-web voor, waar alle beschikbare informatie door een machine leesbaar en semantisch gestructureerd is. Deze semantische structuur is ook gedefinieerd door middels een ontologie, zodanig dat het machines mogelijk is om semantische informatie uit het web te verzamelen. Om semantische informatie onafhankelijk te verzamelen, moet het voor een machine mogelijk zijn om verschillende ontologieën automatisch te mappen. Met behulp van een mapping is het voor een machine mogelijk om informatie te verzamelen welke in een verschillende semantische structuur is gemodelleerd.

Sinds het afgelopen decennium is ontologie-mapping een actief onderzoeksveld. Er zijn gespecialiseerde mapping-systemen ontwikkeld die gebruik maken van een combinatie van technieken om de overeenkomsten tussen concepten te bepalen. Met behulp van deze systemen worden overeenkomende concepten geëxtraheerd, welke vervolgens de alignment tussen de twee ontologieën vormen. In sommige gevallen is het mogelijk dat er extra informatie beschikbaar is, welke gebruikt kan worden om het mapping-proces te verbeteren. Een voorbeeld van dit soort extra informatie is *Wikipedia*. Een lexicale bron zoals Wikipedia maakt het mogelijk om definities van woorden te raadplegen, synonieme woorden te identificeren en informatie van gerelateerde concepten te raadplegen. Een ander voorbeeld van een extra informatiebron is een partiële mapping. Een partiële mapping is een onvolledige mapping welke het resultaat is van een eerdere poging om een mapping tussen de ontologieën te creëren. Deze mapping is onvolledig omdat bijvoorbeeld een domainexpert niet in staat was deze te voltooien wegens tijdgebrek. Een belangrijke uitdaging in het veld van ontologie-mapping is dus het creëren van technieken welke van dit soort informatiebronnen gebruik maken om een mapping te genereren. Dit heeft ons naar de volgende probleemstelling geleid:

> *Hoe kunnen we ontologie-mapping systemen verbeteren door gebruik te maken van externe informatiebronnen?*

Om deze probleemstelling aan te pakken hebben we vier onderzoeksvragen geformuleerd welke dit onderzoek gestuurd hebben:

1. *Hoe kan men nauwkeurig lexicale betekenissen aan ontologieconcepten koppelen?*

2. *Hoe kan men partiële mappings gebruiken om de overeenkomsten tussen concepten te bepalen?*

3. *Hoe kan men beoordelen of correspondenties afkomstig uit partiële mappings betrouwbaar zijn?*

4. *In hoeverre is het mogelijk om parti´ele overeenkomsten te gebruiken om mappings tussen ontologieLen te genereren die weinig overeenkomstige terminologieën gebruiken?*

In hoofdstuk 1 introduceren we het onderzoeksveld van ontologie-mapping. We introduceren de problemen die kunnen ontstaan als men data tussen informatiesystemen wil uitwisselen. Verder introduceren wij een reeks van reële domeinen waar het gepresenteerde werk van toepassing is, zoals bijvoorbeeld informatie integratie, webdienst-compositie en agentcommunicatie. Wij presenteren ook een kort overzicht van de belangrijkste onderzoeksproblemen met betrekking tot ontologie-mapping. In de laatste sectie van dit hoofdstuk introduceren en bespreken wij de probleemstelling en de onderzoeksvragen van dit proefschrift.

Hoofdstuk 2 maakt de lezer bekend met belangrijke achtergrondinformatie. Hier introduceren wij formeel het probleem van ontologie-mapping. Verder detailleren en illustreren we de meest gebruikte technieken waarmee mappings geëvalueerd kunnen worden. Tot slot introduceren wij een aantal datasets die gebruikt kunnen worden om een ontologie-mapping systeem te evalueren.

Technieken die beschikbaar zijn voor het maken vaan een ontologie-mapping worden in hoofdstuk 3 geïntroduceerd. Hier maken wij de lezer bekend met de opbouw van huidige mapping-systemen en de meest gebruikte technieken. Wij introduceren hier de drie kerntaken welke een mapping-systeem moet uitvoeren, namelijk de overeenkomstberekening, overeenkomstcombinatie en correspondentie-extractie. Voor iedere kerntaak geven wij een overzicht van technieken die voor de gegeven taak van toepassing zijn. Vervolgens geven wij een overzicht van huidige mapping-systemen met een focus op systemen die gebruik maken van externe informatiebronnen.

In hoofdstuk 4 beantwoorden wij de eerste onderzoeksvraag. Het kernprobleem hier betreft het nauwkeurig koppelen van lexicale definities aan de gemodelleerde concepten uit de ontologie. Dit proces staat bekend onder de naam *disambiguatie*. Een voorbeeld van dit proces is het vaststellen dat het concept 'Bank' naar het financiële instituut refereert en niet naar het meubelstuk. Technieken die van lexicale informatiebronnen gebruik maken hebben deze koppelingen nodig om de overeenkomsten tussen concepten te bepalen met behulp van bepaalde algoritmen. Wij pakken deze onderzoeksvraag aan door het introduceren van een op Information-Retrieval-gebaseerde techniek waarmee ontologieconcepten aan lexicale betekenissen gekoppeld kunnen worden. Met behulp van deze techniek zetten wij een disambiguatie-kader op waarmee lexicale bedoelingen gefilterd worden afhankelijk van hun overeenkomstwaarden en een filterstrategie. Wij evalueren vier verschillende filterstrategieën die een lexicale betekenis filteren als ze de bijbehorende overeenkomstwaarde onvoldoende vinden. De filterstrategieën worden geëvalueerd met behulp van drie verschillende lexicale overeenskomstmetrieken. Onze evaluatie heeft laten zien dat het toepassen van onze disambiguatieaanpak de prestaties van alle drie overeenkomstmaten heeft verbeterd. Verder hebben wij het effect van het verzwaren van de termgewichten van de concept-annotaties en betekenis-annotaties onderzocht. Deze evaluatie heeft laten zien dat het verzwaren van termgewichten afhankelijk van hun oorsprong in de ontologie of lexicale informatiebron een groter positief effect heeft op de prestatie dan het toepassen van het veelgebruikte TF-IDF aanpak, afkomstig uit het veld van Information-Retrieval.

Het onderzoek in hoofdstuk 5 behandelt onderzoeksvraag 2 in relatie tot het gebruiken van partiële mappings. Het kernprobleem hier is dat het mapping-systeem toegang tot een onvolledige mapping heeft, ook wel een partiëel-mapping genoemd, en dus de onbekende correspondenties moet bepalen om een volledige mapping te creëren. Hier is het doel om de individuele correspondenties van de partiële mapping, ook wel *ankers* genoemd, te gebruiken om de kwaliteit van de berekende mappings te verbeteren. Om deze vraag te beantwoorden stellen wij een methode voor die is gebaseerd op het vergelijken van concepten door het meten van de overeenkomsten tussen een concept en de gegeven ankers. Aan ieder concept worden de overeenkomstwaarden in een zogenoemd *ankerprofiel* samengevoegd. Twee concepten worden als overeenkomend beschouwd als hun ankerprofielen overeenkomen, d.w.z. dat zij vergelijkbare overeenkomsten hebben met de ankerconcepten. In onze evaluatie hebben wij kunnen vaststellen dat onze aanpak in staat is om prestaties te leveren die vergelijkbaar zijn met de top mapping-systemen in het gebied. Onze aanpak is echter wel afhankelijk van het bestaan van geschikte meta-informatie waarmee concepten met ankers worden vergeleken. Hieruit concluderen wij dat alle soorten van meta-informatie geraadpleegd moeten worden door een combinatie van overeenkomstmaten toe te passen om zeker te zijn dat deze techniek voor alle soorten problemen geschikt is. Tot slot voeren wij een systematisch onderzoek uit om vast te stellen hoe groot de invloed is van de grootte en de correctheid van de partiële mapping op de kwaliteit van de berekende mapping. Hier stellen wij vast dat zowel de grootte als ook de correctheid invloed hebben op de mappingkwaliteit. Verder stellen wij vast dat een vermindering van de correctheid van de partiële mapping een sterkere invloed heeft dan een vermindering van de grootte. Hieruit concluderen wij dat mapping-systemen die van partiële mappings gebruik maken maatregelen moeten nemen om ervoor te zorgen dat partiële mappings uit onbekende bronnen correct zijn.

Onderzoeksvraag 3 is het hoofdthema van hoofdstuk 6. De kernvraag hier is het zeker stellen van de correctheid van een gegeven partiële mapping. Sommige technieken genereren een mapping met behulp van een partiële mapping. Een voorbeeld van zo'n techniek is te zien in hoofdstuk 5. Het is voor deze technieken nodig dat de gegeven partiële mapping zo min mogelijk fouten bevat om er voor te zorgen dat deze technieken adequaat presteren. Om de correctheid van een partiële mapping te evalueren stellen wij een techniek voor die gebruik maakt van feature-evaluatietechnieken, afkomstig uit het veld van machine-learning. Om van een feature-evaluatietechniek gebruik te maken moet men eerst een feature-ruimte definiëren. Een feature-ruimte is een kernconcept uit de wiskunde die een ruimte beschrijft die wordt opgespannen door n verschillende features. Bijvoorbeeld, door gebruik te maken van de features 'hoogte', 'breedte' en 'diepte' kan men een 3-dimensionale feature-ruimte opspannen. Door de bijbehorende waardes van de verschillende objecten in deze ruimte te plotten kan men de verhouding van de objecten zien in verband met hun fysieke locatie, en met behulp van deze data verschillende analyses uitvoeren. In het veld van machine-learning zijn er geen beperkingen wat betreft het soort of aantal van gebruikte features. Het is dus ook mogelijk om features te gebruiken om maten zoals positie, grootte, leeftijd, kosten, type of duur weer te geven. Een kerntaak in het veld van machine-learning is classificatie, waar men een categorie aan een object moet toekennen voor welke de waarde van ieder feature bekend is. Een voorbeeld van zo'n taak is het bepalen of een persoon een betrouwbare debiteur is, afhankelijk van zijn inkomen, soort aanstelling, leeftijd en gezinstoestand. Een classificatiesysteem doet dit door eerst een reeks objecten te analyseren van welke de bijbehorende categorieën al bekend zijn. Feature-evaluatietechnieken helpen de maker van een classificatiesysteem te bepalen hoe zeer een bepaalde feature van nut is m.b.t. de classificatietaak door middel van het analyseren van de al geclassificeerde objecten. Voor onze aanpak benutten wij deze werk-flow om een evaluatiesysteem voor een reeks ankers te creëren. Wij zetten een feature-ruimte op waar elke feature het resultaat van een consistentie-evaluatie tussen een specifiek anker en gegeven correspondentie representeert. Met behulp van feature-evaluatietechnieken evalueren wij de kwaliteit van de features, en dus ook de kwaliteit van de bijbehorende ankers. Om de consistentie-waarden te berekenen, definiëren wij een maat welke gebruik maakt van een basis overeenkomstmetriek. Wij evalueren drie soorten maten als basis overeenkomstmetriek: een syntactische, een profiel en een lexicale metriek. Voor ieder type maat evalueren wij onze aanpak ten opzichte van een basis evaluatie, welke gemaakt is door het directe toepassen van de maat op de ankers. Onze evaluatie heeft laten zien dat voor ieder soort maat onze aanpak betere evaluaties produceert dan de bijbehorende basis evaluatie. Voor de syntactische en lexicale maat hebben wij significante verbetering vast kunnen stellen.

Onderzoeksvraag 4 is het onderwerp van het onderzoek dat is beschreven in hoofdstuk 7. Het hoofdthema hier is een specifieke soort van mapping-taken, namelijk het mappen van ontologieën die verschillende terminologieën gebruiken om dezelfde concepten te modelleren. Veel technieken vereisen het bestaan van gelijke of soortgelijke termen om te bepalen of twee concepten met elkaar overeen komen of niet. Deze technieken presteren slecht als de gegeven ontologieën terminologieën gebruiken die doorgaans verschillend zijn. Bestaande technieken vermijden dit probleem door middel van het toevoegen van nieuwe terminologie aan de conceptdefinities. Nieuwe termen worden opgezocht door bijvoorbeeld het raadplegen van bronnen zoals *WordNet*, *Wikipedia* of *Google*. Als dit soort bronnen niet beschikbaar zijn, dan wordt het aanzienlijk moeilijker om een mapping tussen de twee ontologieën te maken. Wij onderzoeken een alternatief dat van een gegeven partiële mapping gebruik maakt. Onze aanpak is gebaseerd op bestaande profiel-overeenkomstmetrieken. Dit soort maten maakt gebruik van semantische relaties om belangrijke contextinformatie te verzamelen. Onze uitbreiding van de gebruikte maat maakt het mogelijk om ook van de semantische relaties in de partiële mapping gebruik te maken. Onze evaluatie heeft laten zien dat onze aanpak in staat is om met behulp van alleen een partiële mapping correspondenties te genereren die een vergelijkbare kwaliteit hebben als bestaande technieken die van lexicale informatiebronnen gebruik maken. Verder stellen wij vast dat een betere kwaliteit haalbaar is als een techniek gebruik maakt van zowel lexicale bronnen als ook partiële mappings.

Hoofdstuk 8 geeft de conclusies van dit proefschrift en bespreekt de mogelijkheden voor verder onderzoek. Rekening houdend met de antwoorden op de gestelde onderzoeksvragen concluderen wij dat er diverse mogelijkheden bestaan om externe informatiebronnen te gebruiken voor het verbeteren van ontologie-mapping-systemen. Ten eerste kunnen lexicale informatiebronnen beter benut worden als eerst een disambiguatie methode toegepast wordt. Ten tweede, door het creëren van ankerprofielen is het mogelijk om overeenkomstwaarden tussen concepten te berekenen. Ten derde, door gebruik te maken van een op feature-selectie-gebaseerde techniek is het mogelijk om ankers te evalueren en er zo voor te zorgen dat technieken die van partiële mappings gebruik maken presteren zoals kan woorden verwacht. Ten vierde, door het uitbreiden van profiel-overeenkomstmetrieken, zodanig dat deze van gegeven ankers gebruik maken, is het mogelijk om ontologieën te mappen die weinig terminologie met elkaar gemeen hebben.

In de discussie over verder onderzoek stellen wij verschillende onderwerpen vast die onderzocht moeten worden om de toepasbaarheid van het gepresenteerde onderzoek te verbeteren. Een onderwerp voor verdergaand onderzoek is dat de robuustheid van de technieken onderzocht moet worden. Bijvoorbeeld, de prestatie van de disambiguatie-techniek van hoofdstuk 4 hangt af van de aanwezigheid van terminologische informatie om zo de correcte lexicale bedoelingen te identificeren. Bij ontologieën waar weinig tot helemaal geen terminologische informatie is gemodelleerd zou het mogelijk kunnen zijn dat de prestaties van de aanpak slechter uitvallen dan verwacht. Een mogelijke oplossing zou het combineren van verschillende disambiguatie-technieken kunnen zijn. Afhankelijk van de beschikbare meta-informatie zou een beslissysteem kunnen bepalen welke disambiguatie-techniek het meest geschikt is voor elke ontologie. Een ander gebied voor verder onderzoek zou het generen van partiële mappings kunnen zijn. Voor problemen waar geen partiële mappping beschikbaar is, zou het genereren van een betrouwbare partiële mapping het mogelijk maken om technieken die gebruik maken van deze partiële mappings toe te passen. Dit zou het dus mogelijk maken om de gepresenteerde technieken op een grotere reeks problemen toe te passen.

About the Author

Frederik C. (Christiaan) Schadd was born on June 16th, 1985 in Hamm, Germany. He received his high-school diploma (Abitur) at the Gymnasium am Turmhof, Mechernich, Germany in 2004. In the same year, he started his bachelor studies in Knowledge Engineering at Maastricht University. During this study he visited Baylor University in Waco, Texas, USA to follow the FastTrac Entrepreneurial Training Program. Frederik received his B.Sc. degree in Knowledge Engineering at the end of the 2007-2008 academic year. He continued his studies at Maastricht University by pursuing a Master's degree in Artificial Intelligence in 2008.

During his studies he also attended an Engagement Workshop in Robotics, Animatronics and Artificial Intelligence in Bristol, UK, and worked as a tutor for the University College Maastricht. In 2010, Frederik completed his Master's degree with the distinction *cum laude*. In the same year, Frederik was employed at Maastricht Instruments as a Software Engineer and Mathematician, where he worked on projects involving medical data analysis and image reconstruction. At the end of this year, Frederik started a Ph.D. project at the Department of Knowledge Engineering at Maastricht University, which resulted in several publications in scientific conferences or journals and ultimately this dissertation. Next to his scientific duties, Frederik was also involved in teaching at the Department of Knowledge Engineering and part of the WebCie commission of the MaasSAC climbing association.

SIKS Dissertation Series

2009

1 Rasa Jurgelenaite (RUN) *Symmetric Causal Independence Models*

2 Willem Robert van Hage (VU) *Evaluating Ontology-Alignment Techniques*

3 Hans Stol (UvT) *A Framework for Evidence-based Policy Making Using IT*

4 Josephine Nabukenya (RUN) *Improving the Quality of Organisational Policy Making using Collaboration Engineering*

5 Sietse Overbeek (RUN) *Bridging Supply and Demand for Knowledge Intensive Tasks - Based on Knowledge, Cognition, and Quality*

6 Muhammad Subianto (UU) *Understanding Classification*

7 Ronald Poppe (UT) *Discriminative Vision-Based Recovery and Recognition of Human Motion*

8 Volker Nannen (VU) *Evolutionary Agent-Based Policy Analysis in Dynamic Environments*

9 Benjamin Kanagwa (RUN) *Design, Discovery and Construction of Service-oriented Systems*

10 Jan Wielemaker (UvA) *Logic programming for knowledge-intensive interactive applications*

11 Alexander Boer (UvA) *Legal Theory, Sources of Law & the Semantic Web*

12 Peter Massuthe (TU/e, Humboldt-Universtät zu Berlin) *Operating Guidelines for Services*

13 Steven de Jong (UM) *Fairness in Multi-Agent Systems*

14 Maksym Korotkiy (VU) *From ontology-enabled services to service-enabled ontologies (making ontologies work in e-science with ONTO-SOA)*

15 Rinke Hoekstra (UvA) *Ontology Representation - Design Patterns and Ontologies that Make Sense*

16 Fritz Reul (UvT) *New Architectures in Computer Chess*

17 Laurens van der Maaten (UvT) *Feature Extraction from Visual Data*

18 Fabian Groffen (CWI) *Armada, An Evolving Database System*

19 Valentin Robu (CWI) *Modeling Preferences, Strategic Reasoning and Collaboration in Agent-Mediated Electronic Markets*

20 Bob van der Vecht (UU) *Adjustable Autonomy: Controling Influences on Decision Making*

21 Stijn Vanderlooy (UM) *Ranking and Reliable Classification*

22 Pavel Serdyukov (UT) *Search For Expertise: Going beyond direct evidence*

23 Peter Hofgesang (VU) *Modelling Web Usage in a Changing Environment*

24 Annerieke Heuvelink (VU) *Cognitive Models for Training Simulations*

25 Alex van Ballegooij (CWI) *RAM: Array Database Management through Relational Mapping*

26 Fernando Koch (UU) *An Agent-Based Model for the Development of Intelligent Mobile Services*

27 Christian Glahn (OU) *Contextual Support of social Engagement and Reflection on the Web*

28 Sander Evers (UT) *Sensor Data Management with Probabilistic Models*

29 Stanislav Pokraev (UT) *Model-Driven Semantic Integration of Service-Oriented Applications*

30 Marcin Zukowski (CWI) *Balancing vectorized query execution with bandwidth-optimized storage*

31 Sofiya Katrenko (UvA) *A Closer Look at Learning Relations from Text*

32 Rik Farenhorst and Remco de Boer (VU) *Architectural Knowledge Management: Supporting Architects and Auditors*

33 Khiet Truong (UT) *How Does Real Affect Affect Affect Recognition In Speech?*

34 Inge van de Weerd (UU) *Advancing in Software Product Management: An Incremental Method Engineering Approach*

35 Wouter Koelewijn (UL) *Privacy en Politiegegevens; Over geautomatiseerde normatieve informatie-uitwisseling*

36 Marco Kalz (OU) *Placement Support for Learners in Learning Networks*

37 Hendrik Drachsler (OU) *Navigation Support for Learners in Informal Learning Networks*

38 Riina Vuorikari (OU) *Tags and Self-Organisation: A Metadata Ecology for Learning Resources in a Multilingual Context*

39 Christian Stahl (TU/e, Humboldt-Universtät zu Berlin) *Service Substitution – A Behavioral Approach Based on Petri Nets*

40 Stephan Raaijmakers (UvT) *Multinomial Language Learning: Investigations into the Geometry of Language*

41 Igor Berezhnyy (UvT) *Digital Analysis of Paintings*

42 Toine Bogers (UvT) *Recommender Systems for Social Bookmarking*

43 Virginia Nunes Leal Franqueira (UT) *Finding Multi-step Attacks in Computer Networks using Heuristic Search and Mobile Ambients*

44 Roberto Santana Tapia (UT) *Assessing Business-IT Alignment in Networked Organizations*

45 Jilles Vreeken (UU) *Making Pattern Mining Useful*

46 Loredana Afanasiev (UvA) *Querying XML: Benchmarks and Recursion*

2010

1 Matthijs van Leeuwen (UU) *Patterns that Matter*

2 Ingo Wassink (UT) *Work flows in Life Science*

3 Joost Geurts (CWI) *A Document Engineering Model and Processing Framework for Multimedia documents*

4 Olga Kulyk (UT) *Do You Know What I Know? Situational Awareness of Co-located Teams in Multidisplay Environments*

5 Claudia Hauff (UT) *Predicting the Effectiveness of Queries and Retrieval Systems*

6 Sander Bakkes (UvT) *Rapid Adaptation of Video Game AI*

7 Wim Fikkert (UT) *Gesture interaction at a Distance*

8 Krzysztof Siewicz (UL) *Towards an Improved Regulatory Framework of Free Software. Protecting user freedoms in a world of software communities and eGovernments*

9 Hugo Kielman (UL) *Politiële gegevensverwerking en Privacy, Naar een effectieve waarborging*

10 Rebecca Ong (UL) *Mobile Communication and Protection of Children*

11 Adriaan Ter Mors (TUD) *The world according to MARP: Multi-Agent Route Planning*

12 Susan van den Braak (UU) *Sensemaking software for crime analysis*

13 Gianluigi Folino (RUN) *High Performance Data Mining using Bio-inspired techniques*

14 Sander van Splunter (VU) *Automated Web Service Reconfiguration*

15 Lianne Bodenstaff (UT) *Managing Dependency Relations in Inter-Organizational Models*

16 Sicco Verwer (TUD) *Efficient Identification of Timed Automata, theory and practice*

17 Spyros Kotoulas (VU) *Scalable Discovery of Networked Resources: Algorithms, Infrastructure, Applications*

18 Charlotte Gerritsen (VU) *Caught in the Act: Investigating Crime by Agent-Based Simulation*

19 Henriette Cramer (UvA) *People's Responses to Autonomous and Adaptive Systems*

20 Ivo Swartjes (UT) *Whose Story Is It Anyway? How Improv Informs Agency and Authorship of Emergent Narrative*

21 Harold van Heerde (UT) *Privacy-aware data management by means of data degradation*

22 Michiel Hildebrand (CWI) *End-user Support for Access to Heterogeneous Linked Data*

23 Bas Steunebrink (UU) *The Logical Structure of Emotions*

24 Dmytro Tykhonov (TUD) *Designing Generic and Efficient Negotiation Strategies*

25 Zulfiqar Ali Memon (VU) *Modelling Human-Awareness for Ambient Agents: A Human Mindreading Perspective*

26 Ying Zhang (CWI) *XRPC: Efficient Distributed Query Processing on Heterogeneous XQuery Engines*

27 Marten Voulon (UL) *Automatisch contracteren*

28 Arne Koopman (UU) *Characteristic Relational Patterns*

29 Stratos Idreos (CWI) *Database Cracking: Towards Auto-tuning Database Kernels*

30 Marieke van Erp (UvT) *Accessing Natural History - Discoveries in data cleaning, structuring, and retrieval*

31 Victor de Boer (UvA) *Ontology Enrichment from Heterogeneous Sources on the Web*

32 Marcel Hiel (UvT) *An Adaptive Service Oriented Architecture: Automatically solving Interoperability Problems*

33 Robin Aly (UT) *Modeling Representation Uncertainty in Concept-Based Multimedia Retrieval*

34 Teduh Dirgahayu (UT) *Interaction Design in Service Compositions*

35 Dolf Trieschnigg (UT) *Proof of Concept: Concept-based Biomedical Information Retrieval*

36 Jose Janssen (OU) *Paving the Way for Lifelong Learning; Facilitating competence development through a learning path specification*

37 Niels Lohmann (TU/e) *Correctness of services and their composition*

38 Dirk Fahland (TU/e) *From Scenarios to components*

39 Ghazanfar Farooq Siddiqui (VU) *Integrative modeling of emotions in virtual agents*

40 Mark van Assem (VU) *Converting and Integrating Vocabularies for the Semantic Web*

41 Guillaume Chaslot (UM) *Monte-Carlo Tree Search*

42 Sybren de Kinderen (VU) *Needs-driven service bundling in a multi-supplier setting - the computational e3-service approach*

43 Peter van Kranenburg (UU) *A Computational Approach to Content-Based Retrieval of Folk Song Melodies*

44 Pieter Bellekens (TU/e) *An Approach towards Context-sensitive and User-adapted Access to Heterogeneous Data Sources, Illustrated in the Television Domain*

45 Vasilios Andrikopoulo (UvT) *A theory and model for the evolution of software services*

46 Vincent Pijpers (VU) *e3alignment: Exploring Inter-Organizational Business-ICT Alignment*

47 Chen Li (UT) *Mining Process Model Variants: Challenges, Techniques, Examples*

48 Milan Lovric (EUR) *Behavioral Finance and Agent-Based Artificial Markets*

49 Jahn-Takeshi Saito (UM) *Solving Difficult Game Positions*

50 Bouke Huurnink (UVA) *Search in Audiovisual Broadcast Archives*

51 Alia Khairia Amin (CWI) *Understanding and supporting information seeking tasks in multiple sources*

52 Peter-Paul van Maanen (VU) *Adaptive Support for Human-Computer Teams: Exploring the Use of Cognitive Models of Trust and Attention*

53 Edgar Meij (UVA) *Combining Concepts and Language Models for Information Access*

2011

1 Botond Cseke (RUN) *Variational Algorithms for Bayesian Inference in Latent Gaussian Models*

2 Nick Tinnemeier (UU) *Organizing Agent Organizations. Syntax and Operational Semantics of an Organization-Oriented Programming Language*

3 Jan Martijn van der Werf (TU/e) *Compositional Design and Verification of Component-Based Information Systems*

4 Hado van Hasselt (UU) *Insights in Reinforcement Learning Formal analysis and empirical evaluation of temporal-difference learning algorithms*

5 Base van der Raadt (VU) *Enterprise Architecture Coming of Age - Increasing the Performance of an Emerging Discipline*

6 Yiwen Wang (TU/e) *Semantically-Enhanced Recommendations in Cultural Heritage*

7 Yujia Cao (UT) *Multimodal Information Presentation for High Load Human Computer Interaction*

8 Nieske Vergunst (UU) *BDI-based Generation of Robust Task-Oriented Dialogues*

9 Tim de Jong (OU) *Contextualised Mobile Media for Learning*

10 Bart Bogaert (UvT) *Cloud Content Contention*

11 Dhaval Vyas (UT) *Designing for Awareness: An Experience-focused HCI Perspective*

12 Carmen Bratosin (TU/e) *Grid Architecture for Distributed Process Mining*

13 Xiaoyu Mao (UvT) *Airport under Control. Multiagent Scheduling for Airport Ground Handling*

14 Milan Lovric (EUR) *Behavioral Finance and Agent-Based Artificial Markets*

15 Marijn Koolen (UvA) *The Meaning of Structure: the Value of Link Evidence for Information Retrieval*

16 Maarten Schadd (UM) *Selective Search in Games of Different Complexity*

17 Jiyin He (UVA) *Exploring Topic Structure: Coherence, Diversity and Relatedness*

18 Mark Ponsen (UM) *Strategic Decision-Making in complex games*

19 Ellen Rusman (OU) *The Mind ' s Eye on Personal Profiles*

20 Qing Gu (VU) *Guiding service-oriented software engineering - A view-based approach*

21 Linda Terlouw (TUD) *Modularization and Specification of Service-Oriented Systems*

22 Junte Zhang (UVA) *System Evaluation of Archival Description and Access*

23 Wouter Weerkamp (UVA) *Finding People and their Utterances in Social Media*

24 Herwin van Welbergen (UT) *Behavior Generation for Interpersonal Coordination with Virtual Humans On Specifying, Scheduling and Realizing Multimodal Virtual Human Behavior*

25 Syed Waqar ul Qounain Jaffry (VU) *Analysis and Validation of Models for Trust Dynamics*

26 Matthijs Aart Pontier (VU) *Virtual Agents for Human Communication - Emotion Regulation and Involvement-Distance Trade-Offs in Embodied Conversational Agents and Robots*

27 Aniel Bhulai (VU) *Dynamic website optimization through autonomous management of design patterns*

28 Rianne Kaptein (UVA) *Effective Focused Retrieval by Exploiting Query Context and Document Structure*

29 Faisal Kamiran (TUE) *Discrimination-aware Classification*

30 Egon van den Broek (UT) *Affective Signal Processing (ASP): Unraveling the mystery of emotions*

31 Ludo Waltman (EUR) *Computational and Game-Theoretic Approaches for Modeling Bounded Rationality*

32 Nees-Jan van Eck (EUR) *Methodological Advances in Bibliometric Mapping of Science*

33 Tom van der Weide (UU) *Arguing to Motivate Decisions*

34 Paolo Turrini (UU) *Strategic Reasoning in Interdependence: Logical and Game-theoretical Investigations*

35 Maaike Harbers (UU) *Explaining Agent Behavior in Virtual Training*

36 Erik van der Spek (UU) *Experiments in serious game design: a cognitive approach*

37 Adriana Burlutiu (RUN) *Machine Learning for Pairwise Data, Applications for Preference Learning and Supervised Network Inference*

38 Nyree Lemmens (UM) *Bee-inspired Distributed Optimization*

39 Joost Westra (UU) *Organizing Adaptation using Agents in Serious Games*

40 Viktor Clerc (VU) *Architectural Knowledge Management in Global Software Development*

41 Luan Ibraimi (UT) *Cryptographically Enforced Distributed Data Access Control*

42 Michal Sindlar (UU) *Explaining Behavior through Mental State Attribution*

43 Henk van der Schuur (UU) *Process Improvement through Software Operation Knowledge*

44 Boris Reuderink (UT) *Robust Brain-Computer Interfaces*

45 Herman Stehouwer (UvT) *Statistical Language Models for Alternative Sequence Selection*

46 Beibei Hu (TUD) *Towards Contextualized Information Delivery: A Rule-based Architecture for the Domain of Mobile Police Work*

47 Azizi Bin Ab Aziz (VU) *Exploring Computational Models for Intelligent Support of Persons with Depression*

48 Mark Ter Maat (UT) *Response Selection and Turn-taking for a Sensitive Artificial Listening Agent*

49 Andreea Niculescu (UT) *Conversational interfaces for task-oriented spoken dialogues: design aspects influencing interaction quality*

2012

1 Terry Kakeeto (UvT) *Relationship Marketing for SMEs in Uganda*

2 Muhammad Umair (VU) *Adaptivity, emotion, and Rationality in Human and Ambient Agent Models*

3 Adam Vanya (VU) *Supporting Architecture Evolution by Mining Software Repositories*

4 Jurriaan Souer (UU) *Development of Content Management System-based Web Applications*

5 Marijn Plomp (UU) *Maturing Interorganisational Information Systems*

6 Wolfgang Reinhardt (OU) *Awareness Support for Knowledge Workers in Research Networks*

7 Rianne van Lambalgen (VU) *When the Going Gets Tough: Exploring Agent-based Models of Human Performance under Demanding Conditions*

8 Gerben de Vries (UVA) *Kernel Methods for Vessel Trajectories*

9 Ricardo Neisse (UT) *Trust and Privacy Management Support for Context-Aware Service Platforms*

10 David Smits (TUE) *Towards a Generic Distributed Adaptive Hypermedia Environment*

11 J.C.B. Rantham Prabhakara (TUE) *Process Mining in the Large: Preprocessing, Discovery, and Diagnostics*

12 Kees van der Sluijs (TUE) *Model Driven Design and Data Integration in Semantic Web Information Systems*

13 Suleman Shahid (UvT) *Fun and Face: Exploring non-verbal expressions of emotion during playful interactions*

14 Evgeny Knutov (TUE) *Generic Adaptation Framework for Unifying Adaptive Web-based Systems*

15 Natalie van der Wal (VU) *Social Agents. Agent-Based Modelling of Integrated Internal and Social Dynamics of Cognitive and Affective Processes.*

16 Fiemke Both (VU) *Helping people by understanding them - Ambient Agents supporting task execution and depression treatment*

17 Amal Elgammal (UvT) *Towards a Comprehensive Framework for Business Process Compliance*

18 Eltjo Poort (VU) *Improving Solution Architecting Practices*

19 Helen Schonenberg (TUE) *What's Next? Operational Support for Business Process Execution*

20 Ali Bahramisharif (RUN) *Covert Visual Spatial Attention, a Robust Paradigm for Brain-Computer Interfacing*

21 Roberto Cornacchia (TUD) *Querying Sparse Matrices for Information Retrieval*

22 Thijs Vis (UvT) *Intelligence, politie en veiligheidsdienst: verenigbare grootheden?*

23 Christian Muehl (UT) *Toward Affective Brain-Computer Interfaces: Exploring the Neurophysiology of Affect during Human Media Interaction*

24 Laurens van der Werff (UT) *Evaluation of Noisy Transcripts for Spoken Document Retrieval*

25 Silja Eckartz (UT) *Managing the Business Case Development in Inter-Organizational IT Projects: A Methodology and its Application*

26 Emile de Maat (UVA) *Making Sense of Legal Text*

27 Hayrettin Gurkok (UT) *Mind the Sheep! User Experience Evaluation & Brain-Computer Interface Games*

28 Nancy Pascall (UvT) *Engendering Technology Empowering Women*

29 Almer Tigelaar (UT) *Peer-to-Peer Information Retrieval*

30 Alina Pommeranz (TUD) *Designing Human-Centered Systems for Reflective Decision Making*

31 Emily Bagarukayo (RUN) *A Learning by Construction Approach for Higher Order Cognitive Skills Improvement, Building Capacity and Infrastructure*

32 Wietske Visser (TUD) *Qualitative multi-criteria preference representation and reasoning*

33 Rory Sie (OUN) *Coalitions in Cooperation Networks (COCOON)*

34 Pavol Jancura (RUN) *Evolutionary analysis in PPI networks and applications*

35 Evert Haasdijk (VU) *Never Too Old To Learn – On-line Evolution of Controllers in Swarm- and Modular Robotics*

36 Denis Ssebugwawo (RUN) *Analysis and Evaluation of Collaborative Modeling Processes*

37 Agnes Nakakawa (RUN) *A Collaboration Process for Enterprise Architecture Creation*

38 Selmar Smit (VU) *Parameter Tuning and Scientific Testing in Evolutionary Algorithms*

39 Hassan Fatemi (UT) *Risk-aware design of value and coordination networks*

40 Agus Gunawan (UvT) *Information Access for SMEs in Indonesia*

41 Sebastian Kelle (OU) *Game Design Patterns for Learning*

42 Dominique Verpoorten (OU) *Reflection Amplifiers in self-regulated Learning*

43 Withdrawn ()

44 Anna Tordai (VU) *On Combining Alignment Techniques*

45 Benedikt Kratz (UvT) *A Model and Language for Business-aware Transactions*

46 Simon Carter (UVA) *Exploration and Exploitation of Multilingual Data for Statistical Machine Translation*

47 Manos Tsagkias (UVA) *Mining Social Media: Tracking Content and Predicting Behavior*

48 Jorn Bakker (TUE) *Handling Abrupt Changes in Evolving Time-series Data*

49 Michael Kaisers (UM) *Learning against Learning - Evolutionary dynamics of reinforcement learning algorithms in strategic interactions*

50 Steven van Kervel (TUD) *Ontologogy driven Enterprise Information Systems Engineering*

51 Jeroen de Jong (TUD) *Heuristics in Dynamic Sceduling; a practical framework with a case study in elevator dispatching*

2013

1 Viorel Milea (EUR) *News Analytics for Financial Decision Support*

2 Erietta Liarou (CWI) *MonetDB/DataCell: Leveraging the Column-store Database Technology for Efficient and Scalable Stream Processing*

3 Szymon Klarman (VU) *Reasoning with Contexts in Description Logics*

4 Chetan Yadati (TUD) *Coordinating autonomous planning and scheduling*

5 Dulce Pumareja (UT) *Groupware Requirements Evolutions Patterns*

6 Romulo Goncalves (CWI) *The Data Cyclotron: Juggling Data and Queries for a Data Warehouse Audience*

7 Giel van Lankveld (UvT) *Quantifying Individual Player Differences*

8 Robbert-Jan Merk (VU) *Making enemies: cognitive modeling for opponent agents in fighter pilot simulators*

9 Fabio Gori (RUN) *Metagenomic Data Analysis: Computational Methods and Applications*

10 Jeewanie Jayasinghe Arachchige (UvT) *A Unified Modeling Framework for Service Design.*

11 Evangelos Pournaras (TUD) *Multi-level Reconfigurable Self-organization in Overlay Services*

12 Marian Razavian (VU) *Knowledge-driven Migration to Services*

13 Mohammad Safiri (UT) *Service Tailoring: User-centric creation of integrated IT-based homecare services to support independent living of elderly*

14 Jafar Tanha (UVA) *Ensemble Approaches to Semi-Supervised Learning Learning*

15 Daniel Hennes (UM) *Multiagent Learning - Dynamic Games and Applications*

16 Eric Kok (UU) *Exploring the practical benefits of argumentation in multi-agent deliberation*

17 Koen Kok (VU) *The PowerMatcher: Smart Coordination for the Smart Electricity Grid*

18 Jeroen Janssens (UvT) *Outlier Selection and One-Class Classification*

19 Renze Steenhuizen (TUD) *Coordinated Multi-Agent Planning and Scheduling*

20 Katja Hofmann (UvA) *Fast and Reliable Online Learning to Rank for Information Retrieval*

21 Sander Wubben (UvT) *Text-to-text generation by monolingual machine translation*

22 Tom Claassen (RUN) *Causal Discovery and Logic*

23 Patricio de Alencar Silva (UvT) *Value Activity Monitoring*

24 Haitham Bou Ammar (UM) *Automated Transfer in Reinforcement Learning*

25 Agnieszka Anna Latoszek-Berendsen (UM) *Intention-based Decision Support. A new way of representing and implementing clinical guidelines in a Decision Support System*

26 Alireza Zarghami (UT) *Architectural Support for Dynamic Homecare Service Provisioning*

27 Mohammad Huq (UT) *Inference-based Framework Managing Data Provenance*

28 Frans van der Sluis (UT) *When Complexity becomes Interesting: An Inquiry into the Information eXperience*

29 Iwan de Kok (UT) *Listening Heads*

30 Joyce Nakatumba (TUE) *Resource-Aware Business Process Management: Analysis and Support*

31 Dinh Khoa Nguyen (UvT) *Blueprint Model and Language for Engineering Cloud Applications*

32 Kamakshi Rajagopal (OUN) *Networking For Learning; The role of Networking in a Lifelong Learner's Professional Development*

33 Qi Gao (TUD) *User Modeling and Personalization in the Microblogging Sphere*

34 Kien Tjin-Kam-Jet (UT) *Distributed Deep Web Search*

35 Abdallah El Ali (UvA) *Minimal Mobile Human Computer Interaction*

36 Than Lam Hoang (TUe) *Pattern Mining in Data Streams*

37 Dirk Börner (OUN) *Ambient Learning Displays*

38 Eelco den Heijer (VU) *Autonomous Evolutionary Art*

39 Joop de Jong (TUD) *A Method for Enterprise Ontology based Design of Enterprise Information Systems*

40 Pim Nijssen (UM) *Monte-Carlo Tree Search for Multi-Player Games*

41 Jochem Liem (UVA) *Supporting the Conceptual Modelling of Dynamic Systems: A Knowledge Engineering Perspective on Qualitative Reasoning*

42 Léon Planken (TUD) *Algorithms for Simple Temporal Reasoning*

43 Marc Bron (UVA) *Exploration and Contextualization through Interaction and Concepts*

2014

1 Nicola Barile (UU) *Studies in Learning Monotone Models from Data*

2 Fiona Tuliyano (RUN) *Combining System Dynamics with a Domain Modeling Method*

3 Sergio Raul Duarte Torres (UT) *Information Retrieval for Children: Search Behavior and Solutions*

4 Hanna Jochmann-Mannak (UT) *Websites for children: search strategies and interface design - Three studies on children's search performance and evaluation*

5 Jurriaan van Reijsen (UU) *Knowledge Perspectives on Advancing Dynamic Capability*

6 Damian Tamburri (VU) *Supporting Networked Software Development*

7 Arya Adriansyah (TUE) *Aligning Observed and Modeled Behavior*

8 Samur Araujo (TUD) *Data Integration over Distributed and Heterogeneous Data Endpoints*

9 Philip Jackson (UvT) *Toward Human-Level Artificial Intelligence: Representation and Computation of Meaning in Natural Language*

10 Ivan Salvador Razo Zapata (VU) *Service Value Networks*

11 Janneke van der Zwaan (TUD) *An Empathic Virtual Buddy for Social Support*

12 Willem van Willigen (VU) *Look Ma, No Hands: Aspects of Autonomous Vehicle Control*

13 Arlette van Wissen (VU) *Agent-Based Support for Behavior Change: Models and Applications in Health and Safety Domains*

14 Yangyang Shi (TUD) *Language Models With Meta-information*

15 Natalya Mogles (VU) *Agent-Based Analysis and Support of Human Functioning in Complex Socio-Technical Systems: Applications in Safety and Healthcare*

16 Krystyna Milian (VU) *Supporting trial recruitment and design by automatically interpreting eligibility criteria*

17 Kathrin Dentler (VU) *Computing healthcare quality indicators automatically: Secondary Use of Patient Data and Semantic Interoperability*

18 Mattijs Ghijsen (VU) *Methods and Models for the Design and Study of Dynamic Agent Organizations*

19 Vinicius Ramos (TUE) *Adaptive Hypermedia Courses: Qualitative and Quantitative Evaluation and Tool Support*

20 Mena Habib (UT) *Named Entity Extraction and Disambiguation for Informal Text: The Missing Link*

21 Kassidy Clark (TUD) *Negotiation and Monitoring in Open Environments*

22 Marieke Peeters (UU) *Personalized Educational Games - Developing agent-supported scenario-based training*

23 Eleftherios Sidirourgos (UvA/CWI) *Space Efficient Indexes for the Big Data Era*

24 Davide Ceolin (VU) *Trusting Semi-structured Web Data*

25 Martijn Lappenschaar (RUN) *New network models for the analysis of disease interaction*

26 Tim Baarslag (TUD) *What to Bid and When to Stop*

27 Rui Jorge Almeida (EUR) *Conditional Density Models Integrating Fuzzy and Probabilistic Representations of Uncertainty*

28 Anna Chmielowiec (VU) *Decentralized k-Clique Matching*

29 Jaap Kabbedijk (UU) *Variability in Multi-Tenant Enterprise Software*

30 Peter de Cock (UvT) *Anticipating Criminal Behaviour*

31 Leo van Moergestel (UU) *Agent Technology in Agile Multiparallel Manufacturing and Product Support*

32 Naser Ayat (UvA) *On Entity Resolution in Probabilistic Data*

33 Tesfa Tegegne (RUN) *Service Discovery in eHealth*

34 Christina Manteli (VU) *The Effect of Governance in Global Software Development: Analyzing Transactive Memory Systems.*

35 Joost van Ooijen (UU) *Cognitive Agents in Virtual Worlds: A Middleware Design Approach*

36 Joos Buijs (TUE) *Flexible Evolutionary Algorithms for Mining Structured Process Models*

37 Maral Dadvar (UT) *Experts and Machines United Against Cyberbullying*

38 Danny Plass-Oude Bos (UT) *Making brain-computer interfaces better: improving usability through post-processing.*

39 Jasmina Maric (UvT) *Web Communities, Immigration, and Social Capital*

40 Walter Omona (RUN) *A Framework for Knowledge Management Using ICT in Higher Education*

41 Frederic Hogenboom (EUR) *Automated Detection of Financial Events in News Text*

42 Carsten Eijckhof (CWI/TUD) *Contextual Multidimensional Relevance Models*

43 Kevin Vlaanderen (UU) *Supporting Process Improvement using Method Increments*

44 Paulien Meesters (UvT) *Intelligent Blauw. Met als ondertitel: Intelligence-gestuurde politiezorg in gebiedsgebonden eenheden.*

45 Birgit Schmitz (OUN) *Mobile Games for Learning: A Pattern-Based Approach*

46 Ke Tao (TUD) *Social Web Data Analytics: Relevance, Redundancy, Diversity*

47 Shangsong Liang (UVA) *Fusion and Diversification in Information Retrieval*

2015

1 Niels Netten (UvA) *Machine Learning for Relevance of Information in Crisis Response*

2 Faiza Bukhsh (UvT) *Smart auditing: Innovative Compliance Checking in Customs Controls*

3 Twan van Laarhoven (RUN) *Machine learning for network data*

4 Howard Spoelstra (OUN) *Collaborations in Open Learning Environments*

5 Christoph Bösch (UT) *Cryptographically Enforced Search Pattern Hiding*

6 Farideh Heidari (TUD) *Business Process Quality Computation - Computing Non-Functional Requirements to Improve Business Processes*

7 Maria-Hendrike Peetz (UvA) *Time-Aware Online Reputation Analysis*

8 Jie Jiang (TUD) *Organizational Compliance: An agent-based model for designing and evaluating organizational interactions*

9 Randy Klaassen (UT) *HCI Perspectives on Behavior Change Support Systems*

10 Henry Hermans (OUN) *OpenU: design of an integrated system to support lifelong learning*

11 Yongming Luo (TUE) *Designing algorithms for big graph datasets: A study of computing bisimulation and joins*

12 Julie M. Birkholz (VU) *Modi Operandi of Social Network Dynamics: The Effect of Context on Scientific Collaboration Networks*

13 Giuseppe Procaccianti (VU) *Energy-Efficient Software*

14 Bart van Straalen (UT) *A cognitive approach to modeling bad news conversations*

15 Klaas Andries de Graaf (VU) *Ontology-based Software Architecture Documentation*

16 Changyun Wei (UT) *Cognitive Coordination for Cooperative Multi-Robot Teamwork*

17 André van Cleeff (UT) *Physical and Digital Security Mechanisms: Properties, Combinations and Trade-offs*

18 Holger Pirk (CWI) *Waste Not, Want Not! - Managing Relational Data in Asymmetric Memories*

19 Bernardo Tabuenca (OUN) *Ubiquitous Technology for Lifelong Learners*

20 Loïs Vanhée (UU) *Using Culture and Values to Support Flexible Coordination*

21 Sibren Fetter (OUN) *Using Peer-Support to Expand and Stabilize Online Learning*

22 Zhemin Zhu (UT) *Co-occurrence Rate Networks*

23 Luit Gazendam (VU) *Cataloguer Support in Cultural Heritage*

24 Richard Berendsen (UVA) *Finding People, Papers, and Posts: Vertical Search Algorithms and Evaluation*

25 Steven Woudenberg (UU) *Bayesian Tools for Early Disease Detection*

26 Alexander Hogenboom (EUR) *Sentiment Analysis of Text Guided by Semantics and Structure*

27 Sándor Héman (CWI) *Updating compressed column-stores*

28 Janet Bagorogoza (TiU) *Knowledge Management and High Performance; The Uganda Financial Institutions Model for HPO*

29 Hendrik Baier (UM) *Monte-Carlo Tree Search Enhancements for One-Player and Two-Player Domains*

30 Kiavash Bahreini (OUN) *Real-time Multimodal Emotion Recognition in E-Learning*

31 Yakup Koç (TUD) *On Robustness of Power Grids*

32 Jerome Gard (UL) *Corporate Venture Management in SMEs*

33 Frederik Schadd (UM) *Ontology Mapping with Auxiliary Resources*

Statements

belonging to the thesis

Ontology Mapping with Auxiliary Resources

by Frederik Christiaan Schadd

1. It is beneficial for lexical similarities to include a concept-sense disambiguation procedure (this thesis, Chapter 4).
2. Partial alignments can serve as reference point for concept comparisons (this thesis, Chapter 5).
3. Before being exploited, given partial alignments should be evaluated with respect to their quality (this thesis, Chapter 6).
4. Concept descriptions can be enriched with additional terminology using partial alignments (this thesis, Chapter 7).
5. Fully automated and reliable mapping systems will become a reality sooner if ontologies are engineered with the process of mapping in mind.
6. A true knowledge economy should enable scientists to do their work instead of constantly forcing them to justify their own existence.
7. An oppressive mind fears criticism, a free mind embraces it.
8. Perseverance is only a virtue when paired with the ability to discard that which does not work.
9. Self-administration of Pavlovian conditioning is an untapped source of human productivity.
10. Doing a PhD is like sports-climbing: sometimes you fail due to lack of strength, at other times due to lack of technique. Both are ultimately fixable.
11. The problem of acquiring science funding would fix itself if PhD students would get a Euro for each time they are asked by their friends and family members how their thesis is progressing.
12. Hakuna Matata.