An Unsupervised Ontology Matching System Using Outlier Analysis

Master Thesis

presented by
Alexander Mller
Matriculation Number 1376818

submitted to the
Chair of Information Systems V
Prof. .Dr. Heiko Paulheim
University Mannheim

Mai 2015

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Overcoming the Disparate Data Space

Maybe use Linked Open Data The Story so far as a motivating example, or the smart data initiative of the federal government of Germany

Defining an ontology is not a deterministic task, so different authors will produce different ontologies that capture the same real life task

1.2 Contributions

Chapter 2

The Ontology Matching Problem

- First define basic terms, like ontology, ontology matching process, ontology alignment
- Give a simple motivating example
- Shortly review the state of the art in ontology matching, mostly ensembles of multiple matchers
- Express challenges, and focus on matcher selection and matcher combination
- Define Problem investigated my master thesis

2.1 Definitions

2.1.1 Ontologies

What is an Ontology

In philosophy the term ontology describes the study of being and existence, trying to define categories of things and to discover relationships among them. Computer Science adopted this term for their own needs and consequently for artificial intelligence and web researchers an ontology is a formal model of a domain.([9], [30]).

In literature there exist various definitions for ontologies on different levels, some of which are discussed in [19]. Nevertheless one of the most cited definitions is the one by [18]: ”An ontology is an explicit specification of a conceptualization”. But probably as often as its cited its extended by other definitions like: ”An ontology is an explicit formal specification of a shared conceptualization of a domain of interest”[38] and ”a logical theory which gives and explicit, partial account of a conceptualization” [20]. These two definitions extends the understanding of an ontology in three points. First of all an ontology needs to be formal, so for instance

a textual description is not sufficient.[38] Moreover it needs to cover a specific domain, so that there exist more than one ontology (a key difference to philosophy). And finally a ontology can only partial conceptualize facts of the real world, so it's a simplified abstraction of the reality.[9]

This rather textual definition will be in the following precised by the introduction of the Web Ontology Language (OWL) which is heavily used in the semantic web (TODO cite linked open data the story to far) and most of the datasets provided by the OAEI are in OWL format. [7]

OWL a Ontology Language

There exist a variety of ontology languages, but in this work the Web Ontology Language (OWL) will be used. OWL is based upon the eXtensinble Markup Language, a succressor of the ontology languages DAMl and OIL, extending RDF and RDFS. In this thesis the OWL 2 Standard is used, which is the second iteration of OWL and became an W3C recommendation 2009 [39]. [1]

OWL is an essential part of the linked open data stack, used as a format to express knowledge about the world. OWL exists in two flavors: OWL DL and OWL Full. The main difference lies in the decidability with reasoning with those ontologies. Since reasoning is not in scope of this thesis, the only thing worth mentioning for completeness it that OWL DL is decidable, because it is based on description logic, where OWL Full not decidable. Despite this fact, the main focus at this point lies in the concepts of OWL to conceptualize facts from the real world, by modeling an ontology. Thus is in the following an introduction of basic OWL elements is given, which is based on [2].

TODO Ontology Listing

The key element of OWL is the definition of classes. They represent an abstraction of concepts from the real world, modeled in the ontology. For instance in Listing XXX an example ontology with 4 classes can be seen. Those classes do have properties and are related to each other. The class Person for example defines the set of person, so each instanziation of the class person is a member of a set containing all persons, these members are usually called individuals or instances.

In order to model complex ontologies classes can also have a subclass relationship. So here we can see that in Listing XXX the class Celebrity is a subclass of Person, so it specializes the Person class. In OWL every class existing is a subclass of owl:Thing the most general class, so each member or instance of a class is also member of the set of instances of owl:Thing. For classes further build-in properties can be used, so for instance classes can be said to be equivalent(owl:equivalentClass) or disjoint(owl:disjointClass). More complex constructs are also possible but are not in scope of this short introduction.

[1] In the following the term OWL is used instead of OWL 2, when to some specialties of OWL 2 is referred, it will be explicitly mentioned

Class may need to have properties to capture real world properties correctly. In OWL there exists two different types of properties: Object properties, which relate individuals with each other, so e.g. the properties lives_in of the Person class, and datatype properties that relate individuals to literal values of a certain datatype, e.g name of the Person class. For those properties a specific domain and range can be defined. So for the property lives_in the domain is that a person can only live in a city, but not in another Person, which kind of makes sense.

In addition to those key properties there exits annotation property, which contain more information about a class. A typical annotation property is rdfs:label, which often gives textual description of the given class. Those properties are ignored in OWL DL and thus are part of OWL Full. Properties itself can have some settings. For instance one can define whether there exists other properties with the same meaning or if they are functional, or symmetric and much more. Those properties will not further be mentioned because they are not relevant to the problem solved in this thesis.

The last component in OWL are individuals, as already mentioned they are instances of a specific class and there for belong to the set of all members of a class. An ontology contains individuals when some base facts needs to be modeled which can be a starting point for inferring new knowledge with reasoners, applied to the ontology. Since reasoning is out of scope in this thesis, individuals are simply treating as parts of an ontology.

Out of this summary of the definition above, now a more formal definition based on [14] of an ontology can be inferred, which will be the basis of the following chapters.

Definition 2.1 (Ontology) *An ontology is a tuple* $o = (C, I, R, T, V, \leq, \perp, \in, =)$ *such that:*

C *is the set of classes,*
I *is the set of individuals,*
R *is the set of relations,*
T *is the set of datatypes,*
V *is the set of values,*
with C, I, R, V *being pairwise disjoint,*
$\leq$ *is a relation on* $(C \times C) \cup (R \times R) \cup (T \times T)$ *called specialization,*
$\perp$ *is a relation on* $(C \times C) \cup (R \times R) \cup (T \times T)$ *called exclusion,*
$\in$ *is a relation over* $(I \times C) \cup (V \times T)$ *called instantiation,*
$=$ *is a relation over* $I \times R \times (I \cup U)$

2.1.2 Ontology Matching

In general ontology matching aims to reduce heterogeneity between different ontologies, to overcome disparate data spaces. As mentioned in the introduction designing an ontology in not a deterministic process (TODO Cite), as heterogeneity

may occur due to different Reasons: For instance because of a different usage pf terms to describe the same real world concept (e.g.: car vs. automobile) or different perspectives on the modeling domain (TODO cite).

Despite that common ground it needs to be said that in literature there exist various terms and definitions for ontology matching. Terms like ontology alignment, ontology mapping, integration of ontologies (TODO cite) are referring to the same concepts and techniques. To have a common understanding in this thesis, the term ontology matching is used when to the process of matching ontologies is referred. Complementary ontology alignment or simply alignment is used when to the resulting output of a matching process is referred. Therefore in the following sections these two questions will be answered:

1. What is a matching process?
2. What is an alignment?

Ontology Matching Process

The foundation for each ontology matching system is a process that matches two or more ontologies in order to produce an alignment between those ontologies. That is called in general a ontology matching process. In the definition of a process is given by considering two input ontologies that should be aligned.

In the database oriented research on ontology matching this process is often called match operator ([34]), which refers in its core to the same abstraction as made in [14] and [9].

Nevermind the fact that matters is that these definitions have in common that the process has as an input of two ontologies that needs to be matched and some parameters to control the internal behavior of the underlying algorithm, usually threshold parameters. In some definitions ([13]) furthermore other resources are also included into the input domain, those can include initial alignments that will bootstrap the alignment process or background knowledge items. The output is consequently an alignment, which consists out of correspondences between entities of the given ontologies. In [14] and [9] the functional character of the matching process is stressed leading to the definition of the matching process.

Definition 2.2 (Matching Process) *An ontology matching process is a function, based on two ontologies to match O_1 and o_2, a set of parameters p and a set of resources r*

$$A = f(o_1, o_2, p, r)$$

TODO Think about adding figure

Considering this function, the expected alignment is a real subset of $A \subseteq o_1 \times o_2 \times \Theta$ with Θ being the set of possible relation types (see Section 2.1.2). This shows that the possible search space can become very big. Assuming that each Ontology contains 1000 entities meaning classes, object properties and data type

properties and possibly 4 type of relations in Θ, the search space is $1000 \times 1000 \times 4 = 4,000,000$. These sizes are rather small especially in the medicine domain exists ontologies much larger than this. Nevermind this shows that the process to find an alignment for ontologies is not trivial. [9] Which properties an alignment has is illustrated in the following section.

Ontology Alignment

The definition of alignments in literature have the following criteria, to express the correspondence between entities, that belong two different ontologies:

1. How to correspond to entities of the underlying ontology
2. Which types of relationships between entities are distinguished
3. How to address the allowed or wished multiplicity of alignments

To tackle the first feature in [14] an entity language is introduced, that defines a separate ontology-format-independent language to address entities and perform operations with them. The basic advantage of this language is that it is a abstraction over the underlying ontology description language and by this allows matching entities of ontology across multiple formats. Since this is not in scope for this work, entities are assumed to have a unique identifier, which they are referenced with, without the use of an entity language.

Another important aspect of a correspondence between two entities is the type of relationship they stand to each other. Those types are often inferred from data modeling techniques([34]) or have set-theoretic background ([14]) and can define for example when two entities are equal to each other ($=$) or one entities more general than another one ($>$). The set of possible relations is called Θ. The predominant relationship in this work is the equality between two entities.

Furthermore in contrast to [14] - but in agreement with [9] and [34]- it is considered that each correspondence has a degree of confidence , expresses the likelihood that the correspondence holds. This confidence is real valued number in the interval $[0, 1]$, where 1 expresses the highest confidence and 0 the lowest. The reason why [14] is not considering a degree of confidence as part of the correspondence definition is that they define a separate meta-data element for each correspondence, which can contain arbitrary information, including the confidence. This work does not follow this definition, because of the high importance of the confidence value for the presented approach to ontology matching.

From all this, the definition of a correspondence and an alignment can be expressed as follows:

Definition 2.3 (Correspondence) *Given two ontologies o_1, o_2 and a set of relations Θ a correspondence is a 4-Tuple:*

$$(e_1, e_2, r, c)$$

with

$$\begin{aligned} e_1 &\in o_1 \wedge e_2 \in o_2 \\ r &\in \Theta \\ c &\in [0,1] \end{aligned}$$

Definition 2.4 (Alignment) *Given two ontologies o_1, o_2 an alignment between them is defined as a set of correspondences of entities e_1 and e_2, where $e_1 \in o_1 \wedge e_2 \in o_2$.*

Ontology Alignment Multiplicity

The previous defined alignment does not consider the allowed multiplicities. But for a lot of problems an alignments with a specific cardinality are necessary. For instance that for each entity of a ontology o_1 needs to be exactly one entity of ontology o_2 mapped, or more likely at most one entity of 0_2.

In [34] and [9] those cardinalities are considered in a way that is known from data modeling languages like the Entity Relationship Diagrams or the Unified Modeling Language in the form of 1:1, 1:N, N:1 and N:M cardinalities between entities of two sets. In [14] however the multiplicities are defined more formally and in a more granular way. Thus we will follow this definition, since this definitions are used in the remainder of this thesis.

In order to define the multiplicity between entities of an alignment, the alignment between two ontologies is treated as a function f where ontology 0_1 is the domain X of f and o_2 is the co-domain Y. Thus we can use the mathematical terms surjective, injective and bijective to define the multiplicity of a function.

A function is surjective when every element $y \in Y$ of f has a corresponding value $x \in X$. This means that the function may map more than one x-value to a y value.

In contrast to that a function is injective when it preserves the uniqueness of a value $x \in X$ when it's mapped to a value $y \in Y$. Therefore intuitively speaking a function is injective if for each value $y \in Y$ it maps at most one value $x \in X$. Combining those two properties a function is bijective if it is surjective and bijective, so for each value of Y has exactly one value in X. In practice this type of function is often called one-to-one mapping.

In [14] however they define one more property of an alignment which is called total. It is an inversion of the surjective properties so that a function is total if for each value $x \in X$ at least one value $y \in Y$ is mapped. Analog to [14] the properties total and injective of an alignment can be defined:

Definition 2.5 (Total and Injective Alignment) *Given two ontologies o_1 and o_2, an alignment A is called total iff:*

$$\forall e_1 \in o_1, \exists e_2 \in o_2 : (e_1, e_2, =) \in A$$

And an alignment A, with the domain o_1 and the codomain 0_2 is called injective from 0_1 to 0_2 iff:

$$\forall e_2 \in o_2, \exists e_{2'}, e_1 \in o_2 : (e_1, e_2, =) \in A \wedge (e_{2'}, e_2, =) \in A \Rightarrow e_1 = e_{2'}$$

In [12] a expressive notation for the definition of multiplicities of alignments in ontology matching is introduced. There the an total and injective alignment is represented by 1, ? represents an injective alignment, + for total and * for an alignment that does not hold the definitions above. These properties are sensitive to the direction they are seen, for instance and A alignment of o_1 and o_2 can be injective from 0_1 to o_2 and total from o_2 to o_1. Thus this results in the following different combinations: ?:?, ?:1, 1:?, 1:1, ?:+, +:?, 1:+, +:1, +:+, ?:*, *:?, 1:*, *:1, +:*, *:+, *:* .

An example for this can be seen in Section 2.2.

2.2 Motivating Example

TODO

2.3 State-of-the Art

There exits several state-of-the-art of the ontology matching systems, developed all over the world. To evaluate the performance of different systems and approaches the Ontology Alignment Evaluation Initiative was started, to asses weaknesses and strength of ontology matching algorithms and by this give developers and researches and developers in this are a platform for knowledge transfer. [11] There the OAEI publishes each year a report on an assesment of current ontology matching systems. In order to validate the state-of-the-art, analyzing the techniques used by participants of OAEI is a good starting point. In this section this finding are shortly summarized, based on those reports. [**?**] [**?**] [**?**] [10] Add other reports

Mostly all systems that attended the challenge have in common that they rely on multiple ontology matching functions. In order to overcome weakness of different types of functions, state-of-the art ontology matching systems use different measures. They exploit the structural properties, the name of elements and sometimes individuals of ontologies in order to match entities. The main drawback of this multiple strategy or hybrid matching is that there is a need of a combination function of the results of the so called base matchers. This can be a weighted average sum (TODO cite YAM) or a majority vote of the base matchers ([8]). [36] Another similarity is modern matchers try to be as efficient as possible and therefore align ontologies faster and in addition be able to align big ontologies.[33] Furthermore most matching approaches use base matchers that rely on background

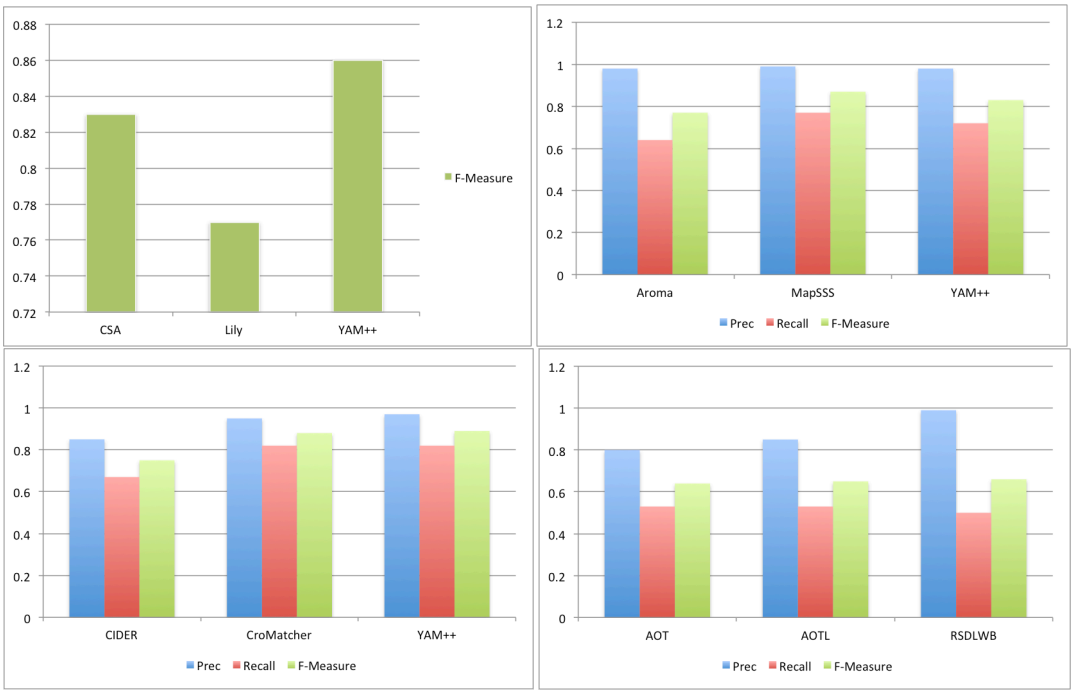

Figure 2.1: Top performing Ontology Matching Systems for the OAEI Bibliography Benchmark Dataset

knowledge, which is used to overcome a terminological mismatch, between the matched ontologies. Such resources can be WordNet [17] or UMLS[3]. [11]

Of particular interest is consequently the performance of matchings systems, which is usually estimated using the measures $rrecision$, $recall$ and $f_1 - measure$ [2] from information retrieval. The OAEI offers different tracks that contain different ontologies to be matched. To illustrate the performance over those different datasets in the following the results of matching systems for four different datasets over last four years will be presented, which is an update of the work done in [36].

2.3.1 Bibliography Benchmark

2.3.2 Conference

This data set contains 21 ontologies from the conference domain. They were either created manually, were present or created based on a website of the conference. The ontologies did not change over time and also the alignments between all ontologies are stable. In figure 2.2 the result of the top performing matching systems are presented. It can be observed that their has been no real improvement in terms of $f_1 - measure$ since 2012. The best systems are AML [16], YAM++ [27] and LogMap [23].

[2]TODO add reference to section

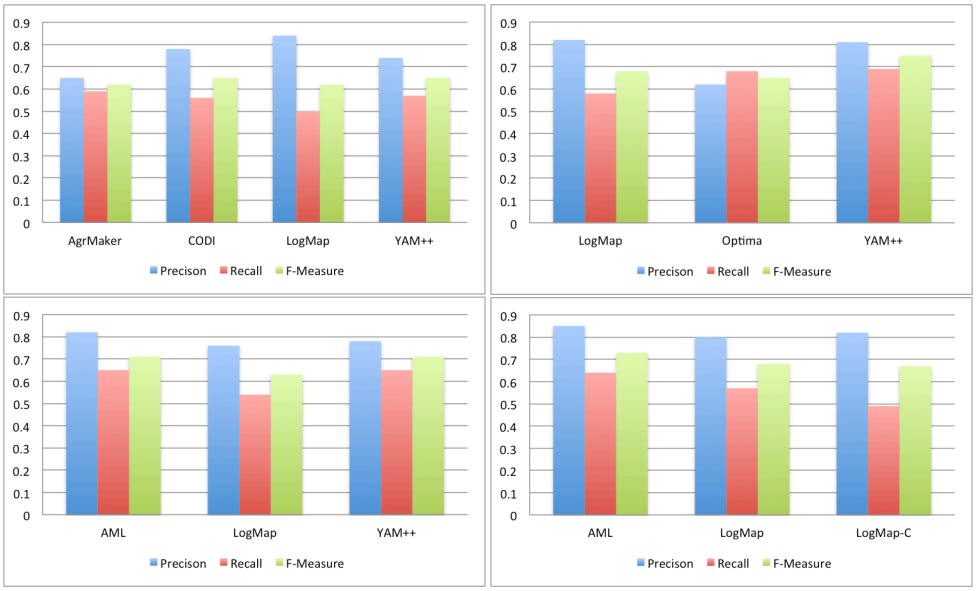

Figure 2.2: Top performing Ontology Matching Systems for the OAEI Conference Dataset

2.3.3 Anatomy

For the anatomy dataset the task is set to match the Adult Mouse Anatomy with NCI Thesaurus [3], that describes the human anatomy. The matching task is rather large and results in a big alignment. Despite some changes in the dataset from 2011 to 2012, it was stable over the last two years and therefore comparisons are possible. What can be seen is that there is a clear improvement in terms of $f_1 - measure$ over the years, with CODI, AML, GOMMA, YAM++ and LogMap as best performing systems in different periods.

2.3.4 Library

The OAEI organizers included in 2012 the current

[3] http://ncim.nci.nih.gov/ncimbrowser/

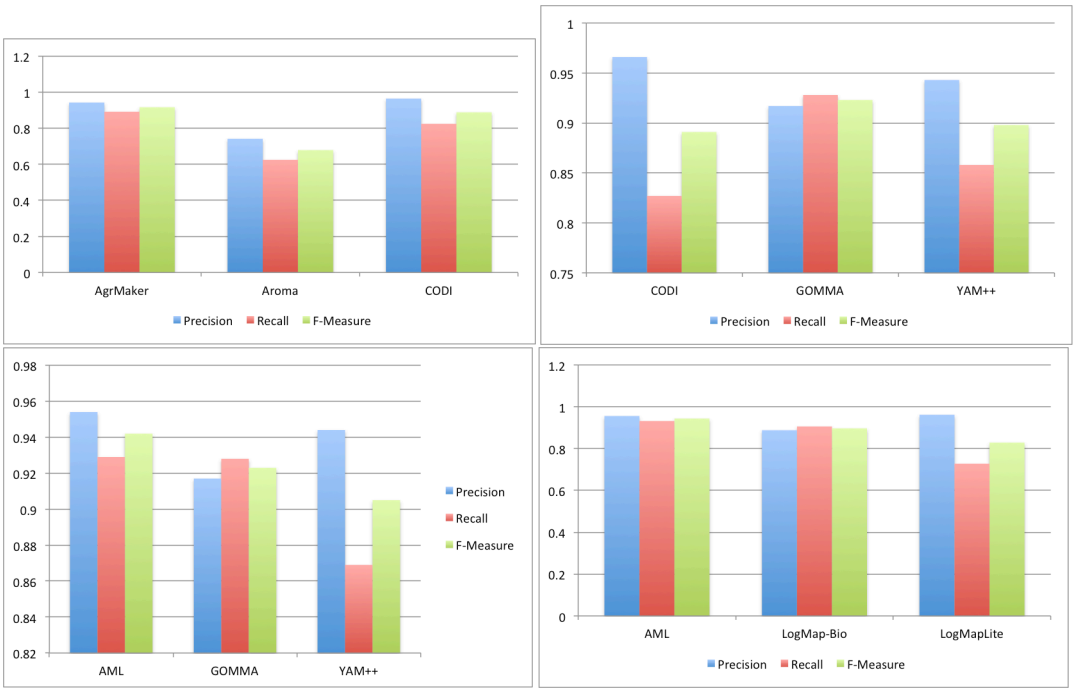

Figure 2.3: Top performing Ontology Matching Systems for the OAEI Anatomy Dataset

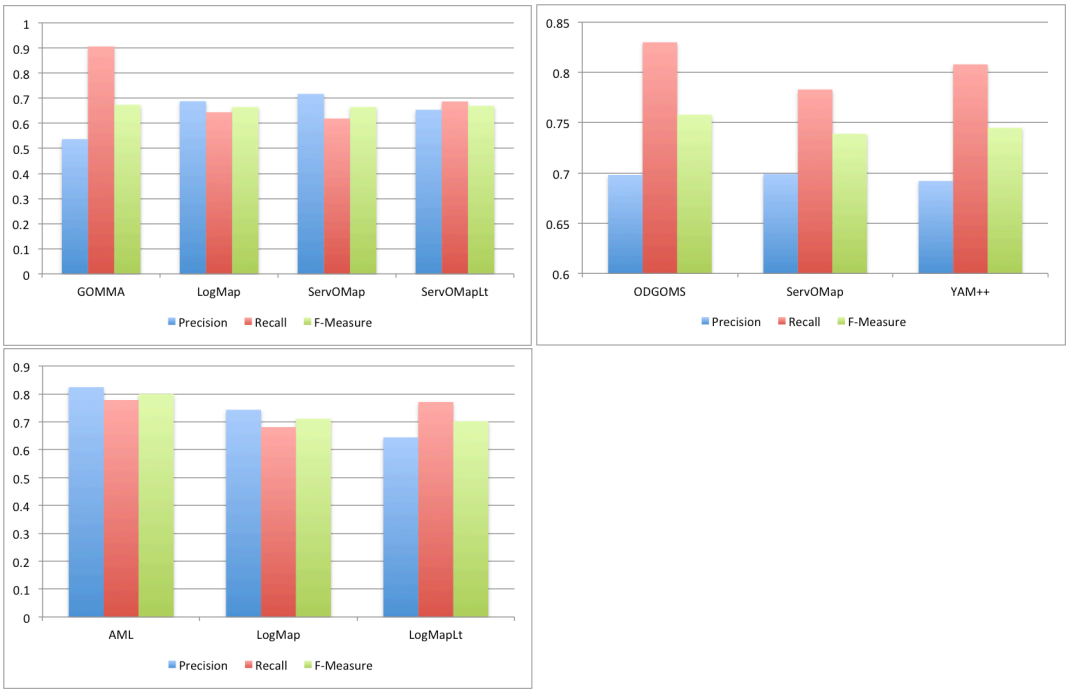

Figure 2.4: Top performing Ontology Matching Systems for the OAEI Library Dataset

2.4 Challenges

The previous section demonstrated that the ontology matching community is improving their systems in terms of $f_1 - measure$. Another way of looking at this is that there still exist various challenges that need be solved, to overcome the disparate dataspace. In [36] and [14] a research agenda with open questions in the field of ontology matching is presented. In addition to that in [29] a survery was performed among 33 researchers in the field of ontology matching. Table 2.1 summarizes the findings.

Future Challenge	Shvaiko & Euzenat	Otero-Cerdeira et. al.
Large Scale and Efficient Matching	✓	✓
Matching with Background Knowledge	✓	✓
Matcher Selection, Combination and Tuning	✓	✓
User Involvement	✓	✓
Explanation of Matching Results	✓	✓
Uncertainty in Ontology Matching	✓	✓
Alignment Management	✓	✓
Improvement of Precision and Recall		✓
Semantic Mapping		✓
Non-expert user tools		✓
Holistic Ontology Matching		✓
Complex Relation Detection (non 1:1)		✓
Practical Applications of created Mappings		✓

Table 2.1: Challenges of Ontology Matching Research base on [29], [14] and [36]

However in this thesis the focus lies on the open challenges of *Matcher Selection, Combination and Tuning*. As already mentioned above current matching systems rely on multiple base matchers that have to be combined to create one final alignment and try by this to overcome the challenges of heterogenous ontology that need to interoperate. These systems need to elect base matchers and finally combine them to obtain a comprehensive result. [5]

The best way of tackling this problem is to divide into two separate problems, first the matcher selection and second the matcher combination problem.

The first problem can be treated as a feature selection problem, where base matchers are potential useful features that need to be selected.

On the other hand there is the second problem which describes the combination of the output of the selected base matcher. Current approaches treat this as a weighted average function of the results of the base matchers. This does not incorporate that for matching specific entities some matchers might not be suitable while others predict the correspondence perfectly correct. Therefore there is a need for finding a flexible way to combine the output of selected base matchers, that embraces strength and punishes weakness of individual matchers, aligning judging

over a specific correspondence. This leads to the definition of when a flexible ontology matching process.

Definition 2.6 (Flexible Ontology Matching Process) *Inspired by [28] in this thesis an ontology matching system is flexible when it automatically selects and combines individual matchers to produce the best possible result in changing data domains.*

To allow a maximum degree of flexibility an unsupervised approach to select and combine multiple matcher will be superior to supervised approaches. Because by this the underlying algorithm does not rely on available training data to match algorithms in a distinct domain. Therefore such an algorithm can used in changing data domains, without additional resources needed. Outlier Detection might a possible methods to perform the combination with the demanded degree of flexibility.

The best way of summing up this section is to formalize the problem that will be investigated in this master thesis. The goal is to find a meta matching process that selects, runs and combines different individual matching processes, by using a matcher selection function f_f and a combination function f_c in the form of:

Definition 2.7 (Meta Ontology Matching Process) *The meta ontology matching process can be seen as a function of two ontologies o_1, o_2, a set of individual matching process F (see definition 2.2), a combination function f_c, a feature selection function f_f, a set of parameter p and a set of resource r, that return an alignment A:*

$$A = f'(o_1, o_1, F, f_c, f_f, p, r)$$

Of particular interest is to find a selection function f_f that relies on unsupervised methods and a combination function f_c that as well has no need for training. Therefore the use of outlier analysis as a flexible unsupervised way will be investigated and evaluated in the remainder of this master thesis.

In this section the ontology matching problem in general was introduced and defined, furthermore the current state-of-the art of ontology matching was based on the OAEI presented. Depended on this analysis some challenges from literature for research in ontology matching were enumerated and especially the problem of individual matcher combination and selection was stressed. Finally the meta ontology matching problem was formalized as finding a flexible matching process for blending multiple results into one final alignment, using outlier detection. Thus in the following section an overview of current individual matching and meta matching techniques is given, in form of an literature review.

Chapter 3

Ontology Matching Approaches (Related Work)

This chapter presents the results of a conducted literature survey in the area of ontology matching.

3.1 Classification of Approaches

Tailored classification of approaches

3.2 Individual Matchers

3.2.1 Element-level Matchers

String-based Matchers

Papers in this area:

- Survey of string metrics used by OAEI Systems in the recent years, based on paper and code of the OAEI Systems [4]
- [14] Chapter 5.2.1
- Standard paper containing, basic methods, all implemented in the second-string library[6]
- String similarity for ontology matching [37] A specificly for ontology matching developed string comparison metric
- Combined levensthein distance with bag of word approache [1]

Language-based Matchers

Rather consider entity names as real language words than simply strings. Preprocessing steps:

- Tokenisation
- Stemming [32]
- lemmatization
- stop word filtering
- bag-of-words representation [35]
- tfidf vector creation [35]

Vector-space-model based:

- TFIDF [35],
- Soft-tfidf [6],

Word Net Based also called in [14] Chapter 5.2.2 extrinsic methods, because they use external resources lexicon, dictionaries and thesauri, therefore they can also be treated also be treated as formal resource-based

- Lin [25]
- Jiang-Conrath [22]
- Wu-Palmer [40]
- Path similarity (The measure path is equal to the inverse of the shortest path length between two concepts) [31]
- WNOntosim [21] TODO write mail for code

3.2.2 Structure-level Matchers

Consider the whole ontology when matching instance and not only the local information present in the entity.Enumerate fields and then say that only graph-based matchers are in scope.

Graph-based Matchers

Explain how to transform the ontology matching problem to a graph matching problem.

Say the graph matching problem here is not a graph matching problem of the global dataspace as proposed in [24].

- Variations of Similarity Flooding [26] e.g. in RiMOM and Yam ++
- Similarity Equation Fixed Point a.k.a OLA [15]
- [24]

3.3 Matching Combination Techniques

3.4 Analysis Combination Techniques

Show weaknesses of current approaches, supervised, often weighted average based, so not flexible (transferable to other domains)

Chapter 4

Selected Approaches to Outlier Analysis (Related Work)

4.1 Definition Outlier Analysis

4.2 Approaches

Chapter 5

An Unsupervised Ontology Matching System Using Outlier Analysis

5.1 Motivation for using Outlier Analysis for Ontology Matching

Flexibility towards changing data domains No need to train weights upfront

5.2 Ontology Matching as an Outlier Detection Problem

5.2.1 Creating the Feature Vector

5.2.2 Significance of Outliers for Ontology Matching

Correlation between Outlier score and label

5.2.3 Transforming the Outlier Analysis Result to a Matching

Chapter 6

A Matching Pipeline using Outlier Detection

6.1 Overview

Presents the implemented Pipeline

6.2 Individual Matchers used

What base matchers survived the selection process

6.3 Feature Selection

6.4 Outlier Analysis to Combine the Results

Chapter 7

Evaluation

7.1 Datasets

7.1.1 Conference

7.1.2 Benchmark

7.1.3 Anatomy

7.1.4 Library

7.2 Experimental Setup

7.3 Used Baselines

7.4 Results

Chapter 8

Discussion

8.1 Flexibility towards changing Data Domains

8.2 Runtime Considerations

8.3 Comparison with current OAEI Participants

Chapter 9

Conclusion

Bibliography

[1] I. Akbari, M. Fathian, and K. Badie. An improved mlma+ and its application in ontology matching. In *Innovative Technologies in Intelligent Systems and Industrial Applications, 2009. CITISIA 2009*, pages 56–60, July 2009.

[2] Grigoris Antoniou, Paul Groth, Frank vanHarmelen, and Rinke Hoekstra. *A Semantic Web Primer*. MIT Press, Cambridge, MA, USA, 2012.

[3] Olivier Bodenreider. The unified medical language system (umls): integrating biomedical terminology. *Nucleic acids research*, 32(suppl 1):D267–D270, 2004.

[4] Michelle Cheatham and Pascal Hitzler. String similarity metrics for ontology alignment. In Harith Alani, Lalana Kagal, Achille Fokoue, Paul Groth, Chris Biemann, JosianeXavier Parreira, Lora Aroyo, Natasha Noy, Chris Welty, and Krzysztof Janowicz, editors, *The Semantic Web â ISWC 2013*, volume 8219 of *Lecture Notes in Computer Science*, pages 294–309. Springer Berlin Heidelberg, 2013.

[5] M Yasser Chuttur. Challenges faced by ontology matching techniques: Case study of the oaei datasets. *Research Journal of Information Technology*, 3(1):33–42, 2011.

[6] William W. Cohen, Pradeep Ravikumar, and Stephen E. Fienberg. A comparison of string metrics for matching names and records. In *KDD WORKSHOP ON DATA CLEANING AND OBJECT CONSOLIDATION*, 2003.

[7] Zlatan Dragisic, Kai Eckert, Jérôme Euzenat, Daniel Faria12, Alfio Ferrara, Roger Granada, Valentina Ivanova, Ernesto Jiménez-Ruiz, Andreas Oskar Kempf, Patrick Lambrix, et al. Results of the ontology alignment evaluation initiative 2014. 2014.

[8] Kai Eckert, Christian Meilicke, and Heiner Stuckenschmidt. Improving ontology matching using meta-level learning. In *Proceedings of the 6th European Semantic Web Conference on The Semantic Web: Research and Applications*, ESWC 2009 Heraklion, pages 158–172, Berlin, Heidelberg, 2009. Springer-Verlag.

[9] M. Ehrig. *Ontology Alignment: Bridging the Semantic Gap*. Semantic Web and Beyond. Springer US, 2006.

[10] J. Euzenat. Todo results of the results of the ontology alignment evaluation initiative 2014. In *Ontology Matching*, 2014.

[11] JÃ©rÃŽme Euzenat, Christian Meilicke, Heiner Stuckenschmidt, Pavel Shvaiko, and CÃ!'ssia Trojahn. Ontology alignment evaluation initiative: Six years of experience. In Stefano Spaccapietra, editor, *Journal on Data Semantics XV*, volume 6720 of *Lecture Notes in Computer Science*, pages 158–192. Springer Berlin Heidelberg, 2011.

[12] Jérôme Euzenat. Towards composing and benchmarking ontology alignments. In *Proc. ISWC-2003 workshop on semantic information integration*, pages 165–166, 2003.

[13] Jérôme Euzenat. An api for ontology alignment. In *The Semantic Web–ISWC 2004*, pages 698–712. Springer, 2004.

[14] Jérôme Euzenat and Pavel Shvaiko. *Ontology matching*. Springer-Verlag, Heidelberg (DE), 2nd edition, 2013.

[15] Jérôme Euzenat and Petko Valtchev. Similarity-based ontology alignment in owl-lite. In Ramon López de Mántaras and Lorenza Saitta, editors, *Proceedings of the 16th Eureopean Conference on Artificial Intelligence, ECAI'2004, including Prestigious Applicants of Intelligent Systems, PAIS 2004, Valencia, Spain, August 22-27, 2004*, pages 333–337. IOS Press, 2004.

[16] Daniel Faria, Catia Pesquita, Emanuel Santos, Matteo Palmonari, IsabelF. Cruz, and FranciscoM. Couto. The agreementmakerlight ontology matching system. In Robert Meersman, HervÃ© Panetto, Tharam Dillon, Johann Eder, Zohra Bellahsene, Norbert Ritter, Pieter De Leenheer, and Deijing Dou, editors, *On the Move to Meaningful Internet Systems: OTM 2013 Conferences*, volume 8185 of *Lecture Notes in Computer Science*, pages 527–541. Springer Berlin Heidelberg, 2013.

[17] Christiane Fellbaum. *WordNet*. Wiley Online Library, 1998.

[18] Thomas R. Gruber. Toward principles for the design of ontologies used for knowledge sharing. *Int. J. Hum.-Comput. Stud.*, 43(5-6):907–928, December 1995.

[19] Nicola Guarino. Understanding, building and using ontologies. *Int. J. Hum.-Comput. Stud.*, 46(2-3):293–310, March 1997.

[20] Nicola Guarino and Pierdaniele Giaretta. Ontologies and knowledge bases: Towards a terminological clarification. In *Towards very Large Knowledge bases: Knowledge Building and Knowledge sharing*, pages 25–32. IOS Press, 1995.

[21] Wei He, Xiaoping Yang, and Dupei Huang. A hybrid approach for measuring semantic similarity between ontologies based on wordnet. In Hui Xiong and W.B. Lee, editors, *Knowledge Science, Engineering and Management*, volume 7091 of *Lecture Notes in Computer Science*, pages 68–78. Springer Berlin Heidelberg, 2011.

[22] Jay J Jiang and David W Conrath. Semantic similarity based on corpus statistics and lexical taxonomy. *arXiv preprint cmp-lg/9709008*, 1997.

[23] Ernesto JimÃ©nez-Ruiz and Bernardo Cuenca Grau. Logmap: Logic-based and scalable ontology matching. In Lora Aroyo, Chris Welty, Harith Alani, Jamie Taylor, Abraham Bernstein, Lalana Kagal, Natasha Noy, and Eva Blomqvist, editors, *The Semantic Web â ISWC 2011*, volume 7031 of *Lecture Notes in Computer Science*, pages 273–288. Springer Berlin Heidelberg, 2011.

[24] Cliff Joslyn, Patrick R. Paulson, and Amanda M. White. Measuring the structural preservation of semantic hierarchy alignment. In Pavel Shvaiko, Jérôme Euzenat, Fausto Giunchiglia, Heiner Stuckenschmidt, Natalya Fridman Noy, and Arnon Rosenthal, editors, *Proceedings of the 4th International Workshop on Ontology Matching (OM-2009) collocated with the 8th International Semantic Web Conference (ISWC-2009) Chantilly, USA, October 25, 2009*, volume 551 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2009.

[25] Dekang Lin. An information-theoretic definition of similarity. In *ICML*, volume 98, pages 296–304, 1998.

[26] Sergey Melnik, Hector Garcia-Molina, and Erhard Rahm. Similarity flooding: A versatile graph matching algorithm and its application to schema matching. In *Proceedings of the 18th International Conference on Data Engineering*, ICDE '02, pages 117–128, Washington, DC, USA, 2002. IEEE Computer Society.

[27] DuyHoa Ngo and Zohra Bellahsene. Yam++ : A multi-strategy based approach for ontology matching task. In Annette ten Teije, Johanna VÃlker, Siegfried Handschuh, Heiner Stuckenschmidt, Mathieu dâAcquin, Andriy Nikolov, Nathalie Aussenac-Gilles, and Nathalie Hernandez, editors, *Knowledge Engineering and Knowledge Management*, volume 7603 of *Lecture Notes in Computer Science*, pages 421–425. Springer Berlin Heidelberg, 2012.

[28] DuyHoa Ngo, Zohra Bellahsene, and Remi Coletta. A flexible system for ontology matching. In Selmin Nurcan, editor, *IS Olympics: Information Systems in a Diverse World*, volume 107 of *Lecture Notes in Business Information Processing*, pages 79–94. Springer Berlin Heidelberg, 2012.

[29] Lorena Otero-Cerdeira, Francisco J. RodrÃ guez MartÃ-nez, and Alma GÃmez-RodrÃ-guez. Ontology matching: A literature review. *Expert Systems with Applications*, 42(2):949 – 971, 2015.

[30] H. Paulheim. *Ontology-based Application Integration*. SpringerLink : Bücher. Springer, 2011.

[31] Ted Pedersen, Siddharth Patwardhan, and Jason Michelizzi. Wordnet:: Similarity: measuring the relatedness of concepts. In *Demonstration Papers at HLT-NAACL 2004*, pages 38–41. Association for Computational Linguistics, 2004.

[32] Martin F Porter. An algorithm for suffix stripping. *Program: electronic library and information systems*, 14(3):130–137, 1980.

[33] Erhard Rahm. Towards large-scale schema and ontology matching. In Zohra Bellahsene, Angela Bonifati, and Erhard Rahm, editors, *Schema Matching and Mapping*, Data-Centric Systems and Applications, pages 3–27. Springer Berlin Heidelberg, 2011.

[34] Erhard Rahm and Philip A. Bernstein. A survey of approaches to automatic schema matching. *The VLDB Journal*, 10(4):334–350, December 2001.

[35] Gerard Salton, A. Wong, and C. S. Yang. A vector space model for automatic indexing. *Commun. ACM*, 18(11):613–620, 1975.

[36] P. Shvaiko and J. Euzenat. Ontology matching: State of the art and future challenges. *Knowledge and Data Engineering, IEEE Transactions on*, 25(1):158–176, Jan 2013.

[37] Giorgos Stoilos, Giorgos Stamou, and Stefanos Kollias. A string metric for ontology alignment. In *Proceedings of the 4th International Conference on The Semantic Web*, ISWC'05, pages 624–637, Berlin, Heidelberg, 2005. Springer-Verlag.

[38] Rudi Studer, V. Richard Benjamins, and Dieter Fensel. Knowledge engineering: Principles and methods. *Data Knowl. Eng.*, 25(1-2):161–197, March 1998.

[39] World Wide Web Consortium (W3C). Owl 2 web ontology language. structural specification and functional-style syntax, 2009.

[40] Zhibiao Wu and Martha Palmer. Verbs semantics and lexical selection. In *Proceedings of the 32nd annual meeting on Association for Computational Linguistics*, pages 133–138. Association for Computational Linguistics, 1994.

Ehrenwörtliche Erklärung

Ich versichere, dass ich die beiliegende Bachelorarbeit ohne Hilfe Dritter und ohne Benutzung anderer als der angegebenen Quellen und Hilfsmittel angefertigt und die den benutzten Quellen wörtlich oder inhaltlich entnommenen Stellen als solche kenntlich gemacht habe. Diese Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen. Ich bin mir bewusst, dass eine falsche Erklärung rechtliche Folgen haben wird.

Mannheim, den 31.5.2015 Unterschrift