

Automatic Schema Matching Utilizing Hypernymy Relations
Extracted From the Web



Master Thesis

presented by
Jan P. Portisch
Matriculation Number 1538843

submitted to the
Data and Web Science Group
Prof. Dr. Heiko Paulheim
University of Mannheim

August 2018

Contents

1	Introduction	1
1.1	Problem Statement	1
1.2	Business Use Case: Semantic Integration in the Financial Services Sector	2
1.3	Pursued Approach	4
2	Theoretical Framework	7
2.1	Semantics	7
2.1.1	General Concepts	7
2.1.2	Semantic Relations	8
2.2	The Semantic Web	10
2.2.1	General Concepts	10
2.2.2	Linked Data	14
2.2.3	The WebIsALOD Data Set	14
2.3	The Ontology Matching Problem	16
2.3.1	General Concepts	17
2.3.2	Ontology Heterogeneity	20
2.3.3	Schema Matching	21
2.3.4	Data Model Schema Matching and Ontology Matching	21
2.3.5	Techniques to Ontology Matching	22
2.3.6	Evaluation of Ontology Alignments	25
2.3.7	Challenges of Ontology and Schema Matching	25
2.4	Natural Language Processing	26
2.4.1	General Concepts	26
2.4.2	Neural Language Models: Word2Vec	28
2.5	Related Work	29
2.5.1	Propositionalization in the Context of Knowledge Graphs	29
2.5.2	Ontology Matching Utilizing External Resources	32

3	Implementation	34
3.1	Matcher Architecture	34
3.1.1	Overview	34
3.1.2	Used APIs and Frameworks	35
3.2	Linking to LOD Resources	37
3.3	Basic Features	41
3.3.1	ALOD Features based on SPARQL	41
3.3.2	Further Features	45
3.4	ALOD2Vec Feature	46
3.4.1	Random Walk Generation	46
3.4.2	Training of Embeddings	50
3.4.3	Hardware	53
3.5	Feature Selection and Weighting	53
3.5.1	Selection and Weighting Process	53
3.5.2	Monosemous Synonymy Gold Standard (MSGS-1234)	53
3.6	Matcher Details	54
3.6.1	Implemented Matching Restrictions	54
3.6.2	Handling of Sub-Concepts	56
3.6.3	Handling of Multiple Textual Parts	57
3.6.4	Performance-Based Enhancements	58
3.7	Implementation Details	60
3.7.1	Projects	60
3.7.2	Code Quality and Documentation	61
3.7.3	Applications	62
4	Experiments	63
4.1	Overview of the Experiments	63
4.2	Descriptive Analysis of the ALOD Data Set	63
4.2.1	Data Set Size	63
4.2.2	Distribution of Relations	64
4.3	Coverage Calculations	67
4.4	Semantic Experiments	69
4.4.1	Gold Standards Used	70
4.4.2	Results	71
4.5	Regressions on Gold Standards	74
4.5.1	Feature Configuration	74
4.5.2	Results	75
4.6	Ontology Matching Experiments	76
4.6.1	Gold Standards Used	77
4.6.2	Features Evaluated	80

4.6.3	Matcher Results	80
4.6.4	OAEI Participation	84
4.6.5	Results Summary	85
4.7	SAP FSDP Matching Use Case	85
4.7.1	Automatic Schema Matching at SAP	85
4.7.2	FSDM Semantic Search	86
4.7.3	Business Value of this Thesis	88
5	Conclusion	89
5.1	Critical Remarks	89
5.1.1	Data Sets	89
5.1.2	Semantic Experiments	90
5.1.3	Ontology Matching and Evaluation Tools	90
5.2	Limits	91
5.2.1	ALOD Data Set	91
5.2.2	Linking to LOD Resources	92
5.2.3	MSGs-1234	93
5.2.4	ALOD Matcher	93
5.3	Challenges for Web-Based Matchers	94
5.4	Summary	94
5.5	Outlook and Future Work	96
A	Program Code / Resources	116
A.1	ALOD Modification Sequence	117
A.2	Python Code: Word2vec Most Related Concepts	118
B	Further Experimental Results	119
B.1	Coverage Statistics: DBpedia vs. ALOD Classic	120
B.2	Similarity Experiments: Correlation of Narrower/Broader Overlap	121
B.3	Similarity Experiments: Correlation of ALOD2Vec	124
C	Further Reference Material	125
C.1	Evaluation Measures	126
C.2	Ten Challenges for Ontology Matching	127
C.3	Paradigmatic Relations	128
C.4	Broader Vector Space Calculation Example	129
C.5	Levenshtein Algorithm	131
C.6	Simlex-999 Instructions	132
C.7	Spearman's Rank Correlation Coefficient	133
C.8	Dice Coefficient	134

C.9 Data Model to Ontology: Transformation Rules	135
C.10 FSDM Semantic Search Server	137
C.10.1 Using the Client on a Windows PC	137
C.10.2 Running the Server (Windows, Linux)	141

List of Figures

1.1	Structure of the Thesis	6
2.1	Two Sides of a Linguistic Sign	8
2.2	Semantic Web Language Stack	11
2.3	RDF Blank Node Example	12
2.4	WebIsALOD Hypernymy Relation Example	16
2.5	Complex Correspondences Example	17
2.6	The Matching Process	19
2.7	Ontology Heterogeneity	20
2.8	Ontology Matching Classification Approaches	23
2.9	Word2Vec Architecture	29
3.1	Matcher Structure Overview	36
3.2	Overlapping Concepts Example	45
4.1	Distribution of Relations of ALOD Classic	65
4.2	Distribution and Relative Share of Relations of ALOD Classic in the Interval [1 – 30]	66
4.3	Distribution of Relations on ALOD XL	66
4.4	Distribution and Relative Share of Relations of ALOD XL in the Interval [1 – 30]	67
4.5	Performance on WordSim	73
4.6	Performance on MEN	74
4.7	Performance on SimLex	75
4.8	FSDM Semantic Search Sample Process	87
C.1	Paradigmatic Relations	128
C.2	ALOD Graph of the Concept <i>Charles Dickens</i>	129
C.3	ALOD Graph of the Concept <i>Ernest Hemingway</i>	130
C.4	Instructions for Simlex-999 Annotators	132

LIST OF FIGURES

C.5	FSDM Semantic Search Welcome Screen	138
C.6	FSDM Semantic Search Process	139
C.7	FSDM Semantic Search Configuration	140
C.8	FSDM Semantic Search Exit	141

List of Tables

3.1	Label Tokenization	40
3.2	ALOD2Vec: Top 10 Closest Concepts	52
3.3	Score Matrix Example	58
4.1	Comparison of ALOD Classic and ALOD XL	64
4.2	Coverage Statistics DBpedia vs. ALOD XL	69
4.3	Absolute Coverage Improvements of ALOD XL over ALOD Classic	70
4.4	Best Spearman’s ρ Values for Overlap and ALOD2Vec	72
4.5	MSGs-1234 Regression Results	76
4.6	OAEI Conference Track Statistics	78
4.7	OAEI Large BioMed Statistics	79
4.8	OAEI Anatomy Matching Results	81
4.9	Examples for True Positives on the Anatomy Data Set	81
4.10	OAEI Conference Matcher Results	82
4.11	OAEI LargeBio Matcher Results	83
4.12	SAP FSDM Matcher Results	84
4.13	Expected Matcher Performance OAEI 2018	85
B.1	Coverage Statistics DBpedia vs. ALOD Classic	120
B.2	Correlation of Narrower/Broader Overlap With WS-353	121
B.3	Correlation of Narrower/Broader Overlap with MEN	122
B.4	Correlation of Narrower/Broader Overlap With SimLex-999	123
B.5	Correlation of ALOD2Vec with WS-353	124
B.6	Correlation of ALOD2Vec with MEN	124
B.7	Correlation of ALOD2Vec with SimLex-999	124
C.1	Contingency Matrix	126
C.2	Vectors of <i>Charles Dickens</i> and <i>Ernest Hemingway</i> in Broader Vector Space	130

Chapter 1

Introduction

1.1 Problem Statement

The amount of data on the Web is constantly growing [83] and with it the embedded knowledge that is publicly available. Utilizing the world's largest public knowledge base is not simple because of the distributed nature of the Web and the subsequent heterogeneity of data such as languages and formats used to represent content [84, p. 10]. However, since the invention of the Semantic Web, the embedded knowledge is much easier to consume in an automated way: The idea behind it is to provide data together with its underlying meaning in a standardized and machine-readable format. The semantics are defined in so called *ontologies*. [78, pp. 10-12] Structured semantic data exploded in recent years which can be seen, for example, when looking at the exponential growth of the number of data sets available in the Linked Open Data Cloud¹.

Ontologies are the first step in allowing interoperability: If the same ontology is used for different applications, those can work together. However, there is not one all-encompassing ontology but multiple ontologies coexist to represent the same kind of information. In order to ultimately use all the available knowledge, one has to mediate between different ontologies in order to allow for interoperability. Therefore, concepts of different ontologies have to be matched which is also known as *ontology alignment* [47, pp. 19-20]. Ideally, this is an automated process which requires little human interaction.

Similar to the Web, the amount of data has also increased in businesses. Data is the foundation of knowledge and the key to increase one's knowledge base [107, p. 58] [48, pp. 1-3]. Today, knowledge is regarded as most valuable resource in

¹see <https://lod-cloud.net/#about>

enterprises [135]. With data being named the *new oil* [168, 169, 160] or the *new gold* [14], the topic of data management and data mining has long ago reached the world of the traditional economy. Nonetheless, the transformation into a "data-driven world" [68] is not easy for such companies as the data is split silo-like into different operational and analytical systems. Each of those systems has its own underlying semantics.

What prevents traditional companies from exploiting the full potential of their data is its heterogeneity. The key to success is overcoming the semantic heterogeneity problem. Mediating the semantics in this context is also known as *data integration* which is a complicated, costly and lengthy process [148, pp. 321-322]. Within this process, ontologies can be used to formally describe the semantics of each data service that is to be integrated. Automatic ontology alignment can then negotiate semantic heterogeneity and, hence, help businesses to integrate their data [49, pp. 8-11].

1.2 Business Use Case: Semantic Integration in the Financial Services Sector

The software landscape of enterprises often resembles a heterogeneous patchwork of different systems by different vendors. Sometimes there are even multiple systems for the same task (e.g. after an acquisition). Reuters reports that Deutsche Bank alone had 45 different IT systems in 2015 which was reduced to 32 as of 2018 [140]. All of those software components use their own data model with a large amount of overlapping parts. For a holistic understanding of the company and its risk profile, all data has to be federated and translated into one view. Especially in the financial sector, an understanding of the company's financial standing as well as its risk exposure is crucial for business decisions. Naturally, there is an endogenous motivation to federate data.

Additionally, regulators emerge to be an exogenous driver for this process by obligating financial institutions to report risk KPIs in a timely manner and even by regulating the IT infrastructure (like BCBS 239² [10]). The costs caused by regulation in the banking sector are considerable. ING Bank, the largest bank of the Netherlands by total assets, for instance, reports in its 2017 annual report EUR 901 million of expenses caused by regulation alone [7, p. 5]. This equals a little less than 10% of the total expenses incurred. The bank describes the situation as fol-

²The Basel Committee on Banking Supervision (BCBS) is a supranational committee which develops regulatory standards for banking institutions. *BCBS* can refer to the committee itself but is also used to refer to its standards (like in BCBS 239) [11].

lows: "Since the start of ECB supervision the increase in regulatory reporting has been significant. Reporting timelines have become shorter and the granularity of the data being requested has increased" [7, p. 17]. From the example of ING Bank, one can see in an exemplary way that there is monetary interest in handling reporting efficiently and that the requirements concerning the IT system are increasing in order to fulfill regulatory standards.

To handle the need of data federation and reporting, all individual data models of different software components have to be reconciled into one holistic view of the company.³ Therefore, links have to be created between the data models in use. Klauck and Stegman's compendium about *Basel III* [98] also covers IT challenges; "integration of source data" as well as "common views on operational and analytical data" are explicitly named [142, pp. 305 - 306]. In another article about semantic integration in the same compendium, ontology mapping is described as the key for semantic integration in banks [116, p. 324].

The required mappings between the data schemas require an enormous amount of manual work to be carried out by well-paid domain experts. Automatic or semi-automatic support during this process can help businesses in tackling the outlined challenges in an efficient way. It gets clear that the problem of ontology and schema matching is not exclusively of academical interest but also relevant for business especially in the financial services domain.

SAP Financial Services Data Platform SAP SE is the world's largest enterprise software company [151, p. 25] and Europe's largest software vendor⁴ [55]. The company is also active in the financial services domain. In December 2017, the company released the *Financial Services Data Platform (FSDP)* to support financial institutions with the management of their data. The product includes a conceptual data model and a physical data model together with an implementation on the SAP HANA database. Eventually, analytical (OLAP) and transactional (OLTP) applications will run on the platform and help financial institutions to simplify their IT landscape. As the product is very young, a lot of applications have to be mapped to the data model of the platform, called *Financial Services Data Management (FSDM) Data Model*. [149, p. 3]

The company also sees the need for the stated problem of ontology matching, is interested in research in this area, and supported this thesis by providing valuable insights, hardware for computations as well as data.

³BCBS 239 explicitly states that "there should be robust automated reconciliation procedures where multiple [data] models are in use" [10, p. 14].

⁴Using sales, profits, assets, and market value as of 2017 in the Forbes 2000 Ranking as benchmark attributes [55].

1.3 Pursued Approach

Ontology Matching "is a solution to the semantic heterogeneity problem" [158, p. 1]. Since the problem is an "AI complete" [43, p. 1] task, the motivation of this master thesis is to facilitate the process by generating high-quality links for general purpose as well as specific data models. Unlike for other mappings, particularly in the financial world simple data type and edit-distance measures are likely not sufficient, domain knowledge is inevitable. In 2008, Shvaiko and Euzenat name "Discovering Missing Background Knowledge" as one of "ten challenges for ontology matching" [156, pp. 1171-1172]. Up to date, the problem is still not solved. The general idea of this thesis is to use publicly available knowledge on the Web for ontology matching to fill the gap of background domain knowledge within the automatic schema matching process. The concepts in focus are often *tail-entities*, i.e., entities interesting only to very few people as opposed to *head-entities* which are more well-known [91, p. 1]. Common knowledge bases of structured information – like Wikipedia-based *DBpedia*⁵ [105] – alone are likely insufficient for this task. Due to the sheer amount of tail-entities such a knowledge base would have to be much larger (Jin et al. assume "an order of magnitude or more entities than in Wikipedia" [91, p. 1]). Tail-entities do exist in large amounts on the Web, however. On Wikipedia, for instance, there is only a general article about future contracts⁶ whereas on the Web, it is possible to find the contract specifications for aluminum alloy futures traded at the London Metal Exchange⁷.

Following this logic, a larger and structured knowledge data set which is based on the whole Web rather than a subset is used in this thesis as background knowledge within the ontology matching process: The WebIsALOD data set [76].

Structure of this Thesis After a quick introduction into the general topic, the following section will focus on the theoretical framework which is based on four pillars: (1) *Semantics* and selected important concepts independent of an IT context are presented in 2.1; (2) *The Semantic Web* is introduced together with the data set which is used in this thesis in section 2.2; (3) *The Ontology Matching Problem* and related concepts in this area are explained in detail in 2.3; and lastly, an overview on (4) *Natural Language Processing* and particularly the *word2vec* approach is given in 2.4. After a sound overview of the fundamental concepts, *Related Work* is presented in 2.5 with a focus on propositionalization and matching with external resources.

⁵see <https://wiki.dbpedia.org/>

⁶see https://en.wikipedia.org/wiki/Futures_contract

⁷see <https://www.lme.com/en-GB/Metals/Non-ferrous/Aluminium-Alloy/Futures>

Subsequent to the theory chapter, the third chapter focuses on the implementation of this thesis. The focus here is on the matcher architecture (3.1), the linking of labels to ALOD concepts (3.2), developed features (3.3 and 3.4) as well as feature weighting methods (3.5). The chapter closes with important calculational details of the matcher (3.6) as well as remarks on the implementation itself (3.7).

Based on the implementation, Chapter 4 describes the experiments that have been performed which are manifold: First, in section 4.2 the data sets used are descriptively analyzed. Furthermore, it is evaluated in how far the WebIsALOD data sets are capable of covering distinct concepts and how that compares to another large linked data set, namely DBpedia, in section 4.3. In addition, experiments were performed to evaluate to which degree the developed features carry semantic meaning and how that compares to other approaches (4.4). Section 4.5 evaluates the feature selection method based on a self-created gold standard and section 4.6 eventually covers the performance of the implemented matchers on multiple commonly used ontology alignment data sets. As the focus of this thesis is also the applicability in the business world, the last section (4.7) spotlights a concrete use case at SAP.

Chapter 5 critically assesses the overall setting as well as the experiments that were performed (5.1) and outlines the limits of the approach presented (5.2). After discussing the challenges for Web-based matchers and particularly the matcher developed in this thesis (5.3), a conclusion (5.4) as well as an outlook (5.5) is given. An overview of the structure of this thesis is presented in figure 1.1.

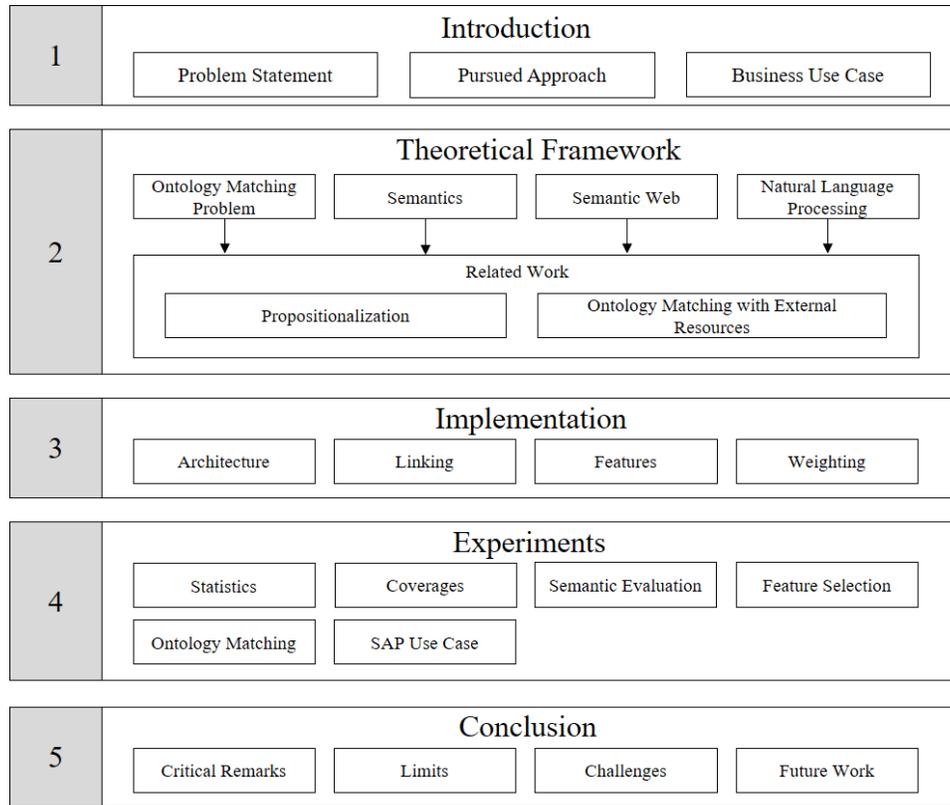


Figure 1.1: Structure of the Thesis

Chapter 2

Theoretical Framework

2.1 Semantics

In this section, a general introduction into semantics is given and aspects relevant for this thesis are explained: First, the difference between *syntax* and *semantics* is pointed out. Afterwards, important relations between concepts in the semantic space are introduced.

2.1.1 General Concepts

Syntax On a general level, syntax refers to a set of rules that define how to structure characters and strings [78, p. 13]. In linguistics, it refers to the analysis of the arrangements of words, phrases, and clauses together with their grammatical relations [21, p. 431].

Semantics Semantics is "the study of meaning" [141, p. 6].¹ The meaning of a word can also be referred to as *concept* [104, p. 21]. As the field of semantics is too broad to be presented in the scope of this thesis, the focus in the following lies on a subset, i.e., semantic relations among concepts.²

¹The meaning of *meaning*, i.e., the question of what meaning actually is, is itself an interesting research area which is – due to the focus of this thesis – not covered at this point. For details, one can refer to Riemer who dedicates a full 40 pages long chapter of his textbook *Introducing Semantics* to this topic [141, pp. 45-85].

²A concise introduction into semantics for non-linguists can be found in Busse's book *Semantik* [28].

2.1.2 Semantic Relations

Every linguistic sign (i.e., word or lexeme³) itself is a relation between the signifier (also *sound-image*, French: *signifiant*) and the signified (the concept, French: *signifié*) [152, p. 77-79], as depicted in figure 2.1.

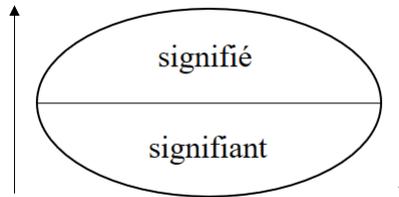


Figure 2.1: Two sides of a Linguistic Sign According to Saussure [152, p. 78]

Besides the relation between signifiant and signifié, there are also relations between signs: *syntagmatic relations* and *paradigmatic relations* (also *associative relations*).

Syntagmatic Relations

Syntagmatic relations are those between signs in a chain of signs; in the English language, for instance, it is grammatically correct to say "he sleeps at night" but not "he sleep at night" because the verb and the subject have to agree in person. [28, p. 102] [34, p. 160]

Paradigmatic Relations

Paradigmatic relations are associations of concepts that exist in the mind of humans but are not necessarily existent in the chain of signs. When reading "to sleep", for instance, there is an implicit association with "sleeping", "bed", "night", and so on. [152, pp. 147-148; 152] [28, p. 102]

Busch and Stenschke count more than ten possible paradigmatic relations (see figure C.3 in the appendix for a complete overview) [27, p. 189]. In the following, only the paradigmatic relations relevant for this thesis are further explained: Hypernymy and Hyponymy, Monosemy and Polysemy, Synonymy and Antonymy, Homonymy as well as Similarity and Relatedness.

³A lexeme is "a unit of lexical meaning, which exists regardless of any inflectional endings[...]" [34, p. 118]. It is also known as *lexical item* [34, p. 464]. The "headwords in a dictionary are [...] lexemes" [34, p. 118].

Hypernymy and Hyponymy A hypernym (also hyperonym) is a concept which is superordinate to other concepts, i.e., it defines a category to which other concepts belong to. Those subordinate concepts are called hyponyms. [27, p. 191] [28, p. 105] The concept of a *financial contract*, for instance, subsumes the concept of a *loan*; therefore, the *financial contract* is a hypernym of *loan* whereas the latter one is a hyponym of the first one.

Monosemy and Polysemy Polysemy describes the property of a lexeme to carry more than one meaning [28, p. 104]. The concept of *apple*, for example, can refer to (i) the fruit, (ii) the tree, or (iii) the Californian technology company; the concept is, therefore, polysemous. A monosemous lexeme, in contrast, carries only one meaning.

Synonymy and Antonymy Synonymy describes the property of two words to be usable interchangeably. Within this definition, there are various forms which mainly focus around whether synonyms have to share one sense, i.e., are interchangeable in one particular context or whether they have to share all senses, i.e., are interchangeable in (almost) all contexts. A strong-form definition of synonymy requires the two words to be interchangeable in any situation. Strong-form synonymous words are seldom. [28, p. 104] An example for weak-form synonymy would be *student* and *pupil* regarding the sense of *somebody being taught by a teacher* but not regarding the sense *being the center of the eye* [141, p. 152]. The words *doorknob* and *doorhandle*, on the other hand, have only one and the same meaning and could be used as an example for strong-form synonymy [133].⁴ Antonyms, on the other hand, are incompatible with each other like *hot* and *cold* [27, p. 191]. If antonyms divide a domain in exactly two parts and are logically incompatible at the same time, like *dead* and *alive*, they are considered to be a *contradiction* [28, p. 106].

Homonymy Words with the same writing and pronunciation but different meaning are called *homonyms* [130, p. 169]. An example for a homonym would be *bear* which – depending on the context – can refer to the animal (*Winnie-the-Pooh is a bear.*) or to the verb (*I cannot bear it any longer.*).

⁴Note that when having a very close look, there are still subtle differences; even though they carry the same sense, *doorhandle*, for example, is more common in Great Britain whereas *doorknob* is mostly used in the United States [133]. This goes even as far as some linguists believing "that there is no such thing as true synonymy" [130, p. 171].

Similarity and Relatedness Similarity describes in how far two concepts are similar to each other "by virtue of their similarity" [25, p. 1]. Similarity and relatedness are often not clearly separated from each other (for instance in [53]). Nevertheless, there are significant differences. Dissimilar entities can even be semantically related by antonymy relationships [25, p. 1]. Hill et al. distinguish the two relations by giving examples: While the concepts *coffee* and *cup* are certainly related, they are not similar; however, a mug and a cup can – in language as in the real world – almost be used interchangeably and are, therefore, similar [77, p. 665]. In this thesis, similarity is treated as the degree of synonymy.

2.2 The Semantic Web

In this section a general introduction into the Semantic Web is given. First, general concepts of the Semantic Web are introduced. Then, linked data is explained and, lastly, the data set used in this thesis is presented.

2.2.1 General Concepts

Semantic Web While information is broadly available on the Web and consumable by humans, computers cannot consume this information due to data heterogeneity and lack of implicit knowledge. One solution would be to have an artificial intelligence that actually can interpret all the information as it is. However, up to now there is no such artificial entity that can reliably accomplish this task. The idea of the *Semantic Web*, on the other hand, is to give information right away in a format that can be interpreted by machines and to provide the required tool set to do so.⁵ The Semantic Web provides standards to ensure interoperability and to allow reasoning according to logic. [78, pp. 9-13] The Semantic Web technology is sometimes also referred to as *Web 3.0* [67, p. 111].⁶

Semantic Web Language Stack In figure 2.2, the Semantic Web language stack is depicted. The technical foundations are Unicode and Uniform Resource Identifiers (URIs). Together, they allow to uniquely identify concepts on the Web in the desired language. The *eXtensible Markup Language* (XML) is a language that allows to exchange structured data in a machine- and human-readable way [170].

⁵Although information can be provided directly so that it is consumable by computers, it is also possible that extractors derive structured information from websites. An example for such a process would be the implementation of *DBpedia* [105] or *DBkWik* [79].

⁶Unfortunately, the term *Web 3.0* is often used for marketing purposes due to the success of the term *Web 2.0*. Therefore, there is no real definition for *Web 3.0* and it is used to refer to different things ranging from virtual worlds [122] to decentralized services such as cryptocurrencies [178].

The *Resource Description Framework* (RDF) allows to express simple statements on the Web [175]; it is further explained in the following paragraph.

RDF Schema (RDF-S) and the Web Ontology Language (OWL) are used to give meaning to the vocabulary used in RDF statements. Rules can additionally be used to express semantics on a deeper level. OWL and RDF-S are explained later in this section in more detail. The *SPARQL Protocol and RDF Query Language* (SPARQL) allows to query RDF data [172].

By combining RDF data and the corresponding semantics, logical inference is possible. This process is referred to as *reasoning*. [47, p. 42] Because "anybody can say anything about anything" (*AAA Principle*) [2, pp. 7-8], there might be multiple views on the truth. Thus, it is valuable to evaluate the credibility of sources and to build trust.

In this thesis, the focus is on the middle layer of the stack, mainly RDF, SPARQL, and OWL. Therefore, selected concepts are explained in more detail in the following.⁷

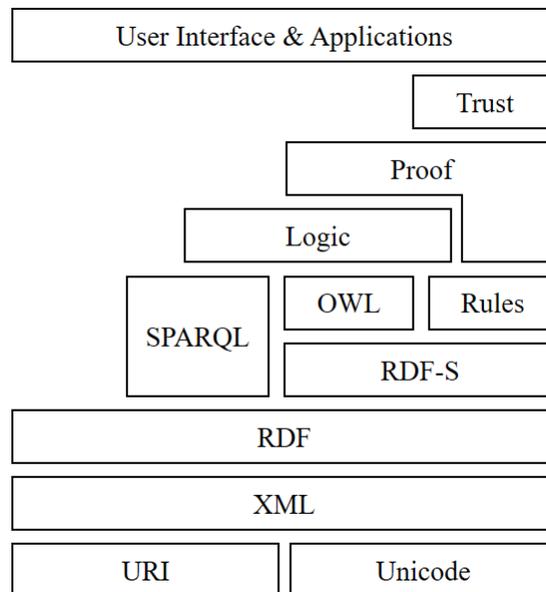


Figure 2.2: Semantic Web Language Stack According to Tim Berners-Lee [16, p. 11] (adapted)

⁷There is more to the Semantic Web than the content described in this section. A comprehensive introduction is given in Hitzler et al.'s textbook *Semantic Web* [78].

Resource Description Framework (RDF) To represent information about resources in a structured form, the W3C developed the *Resource Description Framework* (RDF). The data model behind this standard is relatively simple; statements are given in triples: <subject> <predicate> <object>. Resources are uniquely identified by URIs. When regarding subjects and objects as nodes and predicates as edges, multiple triples can form a connected graph; a very small sub-graph of the ALOD data set is depicted in figure 2.4. This structure allows to interlink knowledge on the Web. [78, pp. 36-39]

In some cases it is necessary to model more complex relations that would require *helper nodes*. An example would be a network of friends where it shall be expressed when people met for the first time. In such cases, blank nodes are used. They are addressed by using a node ID but cannot be addressed by an URI (which would be semantically questionable). An example is given in figure 2.3. [78, pp. 56-58]

For RDF serialization, different formats are available such as Turtle [176] or JSON-LD [173]. There are also formats to serialize multiple graphs in one file such as N-Quads [174].

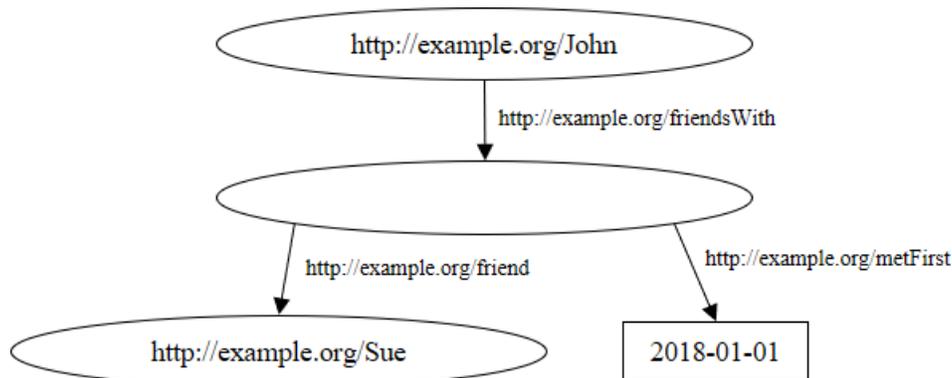


Figure 2.3: RDF Blank Node Example

Ontologies *Ontology*, from Latin *ontologia* derived from Greek *οντος* ('being') and *λογος* ('study of'), is originally a part of philosophy that focuses on the question of *being*, i.e. the nature of the world [26, pp. 4-6] [9, pp. 170-171].⁸ In philosophy, the terms *ontology* and *metaphysics* are often used interchangeably [26, p. 5].

⁸Bunge and Mahner [26] give an excellent (and understandable) introduction into the philosophical dimension of *ontology*.

In information technology, the term *ontology* is used to refer to a specific formalization of concepts: Gruber defines a conceptualization as an "abstract, simplified view of the world" and an ontology as "an explicit specification of a conceptualization" [61, p.1]. In the context of the Semantic Web, an ontology models a domain and defines a vocabulary to be used by an application [49, p. 25]. Two important concepts of ontologies are classes and properties: Classes define the type of a resource whereas properties are the predicates of a statement. Classes and properties can be hierarchically structured, i.e., it is also possible to define sub-classes as well as sub-properties. [78, pp. 68-77]

An example for an ontology would be the *Friend-of-a-Friend* ontology (FOAF)⁹ which can be used to describe social networks, for instance [20]. Ontologies are also already used directly in the business world, for example in the form of industry specific ontologies such as the *Financial Industry Business Ontology (FIBO)*¹⁰ by the EDM Council. Oberle et al. also describe the usage of concrete enterprise applications [124].

An ontology consisting of "meta, generic, abstract and philosophical" [165] concepts is also referred to as *upper ontology* by the IEEE Upper Ontology Working Group¹¹; however, the term is likewise used to refer to ontologies with general concepts (for instance in [114]). An example for an ontology according to a strict definition would be *DOLCE* [56] whereas SUMO [123] or OpenCyc [108] are illustrations for more general purpose upper ontologies containing also domain-specific concepts [114, p. 621].¹²

Ontology Languages There are multiple languages and ways to represent ontologies.¹³ A lightweight format to do so is RDF Schema (RDFS, RDF-S). [78, pp. 66-69] A more expressive and powerful format is the Ontology Web Language (OWL) which is structured in three sub languages listed here in descending expressibility: OWL Full, OWL DL, and OWL Lite. OWL Lite is a subset of OWL DL and OWL DL is a subset of OWL Full. [78, pp. 125-127] OWL is recommended by the W3C [171] and also the language of choice to represent ontologies within the scope of this thesis.

⁹see <http://www.foaf-project.org/>

¹⁰see <https://www.edmcouncil.org/financialbusiness>

¹¹The Upper Ontology Working Group has resolved by now. Nevertheless, their definitions are still available using web.archive.org, see <http://web.archive.org/web/20140512225349/http://suo.ieee.org/>.

¹²*DOLCE* is an acronym for Descriptive Ontology for Linguistic and Cognitive Engineering; *SUMO* is an acronym for *Suggested Upper Merged Ontology* and *OpenCyc* is derived from *Open Encyclopedia*. All three ontologies are rather addressed using their abbreviated forms.

¹³Staab and Studer dedicate more than 100 pages to this topic in their *Handbook on Ontologies* [164, pp. 19-132]

SPARQL Similar to SQL for data bases, the *SPARQL Protocol and RDF Query Language* (SPARQL) allows to query RDF data. Queries are formulated as patterns that are matched against a knowledge graph. In addition, more complex structures such as filters, aggregations, or optional patterns are also available. [78, pp. 202-232] An example for a simple query is given in listing 2.1. Originally designed as pure query language, version 1.1 offers functions to update data [172].

```

1 PREFIX : <http://dbpedia.org/resource/>
2 PREFIX dbo: <http://dbpedia.org/ontology/>
3 SELECT ?population WHERE {
4   :Mannheim dbo:populationTotal ?population .
5 }
```

Listing 2.1: SPARQL Sample Query that will return the population of Mannheim. The query can be run on the public DBpedia endpoint.¹⁵

2.2.2 Linked Data

Tim Berners-Lee defined four principles for linked data which are given word-by-word in the following enumeration [15]:

1. Use URIs as names for things[.]
2. Use HTTP URIs so that people can look up those names.
3. When someone looks up a URI, provide useful information, using the standards (RDF*, SPARQL)[.]
4. Include links to other URIs so that they can discover things.

He further defines Linked Open Data (LOD) in 2010 as "Linked Data which is released under an open license, which does not impede its reuse for free" [15].

2.2.3 The WebIsALOD Data Set

When working with knowledge bases in order to exploit the contained knowledge in applications, a frequent problem is the fact that less common entities are not contained within the knowledge base [153, p. 1]. The *WebIsA* database is an attempt to tackle this problem by providing a data set which is not based on a single source of knowledge – like DBpedia – but instead on the whole Web: The data set

¹⁵see <http://dbpedia.org/snorql/?query=SELECT+%3Fpopulation+WHERE+%7B%0D%0A%09%3AMannheim+dbo%3ApopulationTotal+%3Fpopulation+.%0D%0A%7D>

consists of hypernymy relations extracted from the *Common Crawl*¹⁶, a downloadable copy of the Web [153]. For the extraction, lexico-syntactic patterns similar to those presented by Hearst [66] were used [153, p. 3]. An example for such a pattern would be NP_h such as NP_t where NP_t denotes the hyponym and NP_h denotes the hypernym. From the sentence *omnivores such as rats, rat* would be identified as hyponym and *omnivore* would be identified as the corresponding hypernym (after lemmatization). In total, 58 such patterns were used. [153, p. 5]

WebIsA consists of more than 400 million *isa* relations and 107 million unique hypernyms [153, p. 5]. Altogether, there are 212,155,729 unique concepts available on WebIsA [153, p. 5].¹⁷ To put this number into perspective: The Oxford Dictionary counts 218,632 entries (out of which 47,156 are regarded as obsolete words that are not in current use), roughly half of the entries refer to nouns [126]; the English Wikipedia has a little over 5.6 million articles¹⁸.

In 2017, a Linked Open Data endpoint was added, called *WebIsALOD* [76], which allows to query the data set (including metadata) using SPARQL. Additionally, machine learning was used to assign a confidence score to each relation and links to DBpedia and YAGO were added. The data set of the original endpoint is filtered to ensure a higher data quality [76, p. 113]. Given a specific relation r from the set of all relations R with $|r.pld|$ being the number of pay-level domains (PDL) on which the relation appears and with $|r.pid|$ being the number of pattern IDs that match the relation, the filter is defined as follows [76, pp. 112-113]:

$$dataset(t) = \{r \in R \mid |r.pld| > t \wedge |r.pid| > t\} \quad (2.1)$$

In WebIsALOD, the filter parameter is set to $t = 1$. Hence, all hypernyms in the data set occur in at least two pay-level domains and match at least two patterns. This leads to a significant reduction of the size of the data set (see table 4.2.1 in section 4.2.1). For the reason of simplicity, the data set is also addressed by using the abbreviated form *ALOD* further on.

In 2018, a larger endpoint was added without any filtering: In the following, the smaller endpoint is referred to as *ALOD Classic*¹⁹ and the larger endpoint

¹⁶see <http://commoncrawl.org>

¹⁷This number is not explicitly stated in the paper but can easily be calculated with the numbers given in Table 3 of [153, p. 5]: It is the number of unique concepts that appear in the role of a hyponym ($|\{t_T\}|$) plus the number of unique concepts that appear in the role of a hypernym ($|\{h_T\}|$) minus the intersection of those: $|\{t_T\}| + |\{h_T\}| - |\{t_T\} \cap \{h_T\}|$. Plugging in numbers, one obtains: $120,992,255 + 107,691,822 - 16,528,348 = 212,155,729$. In subsection 4.2.1, detailed statistics about this data set are presented which confirm this number.

¹⁸see: <https://en.wikipedia.org/wiki/Special:Statistics> as of June 2018.

¹⁹Endpoint: <http://webisa.webdatacommons.org/sparql>

is referred to as *ALOD XL*²⁰. An exemplary hypernymy relation and queryable metadata is depicted in figure 2.4.

Remark Throughout this thesis, Internationalized Resource Identifiers (IRIs)²¹ are given when concrete concepts are discussed. Note that while the ALOD Classic instances can be looked at online through a Web interface, the ALOD XL instances cannot be viewed online but have to be queried, e.g., by using the online SPARQL endpoint. XL IRIs are, therefore, marked as such in the following.

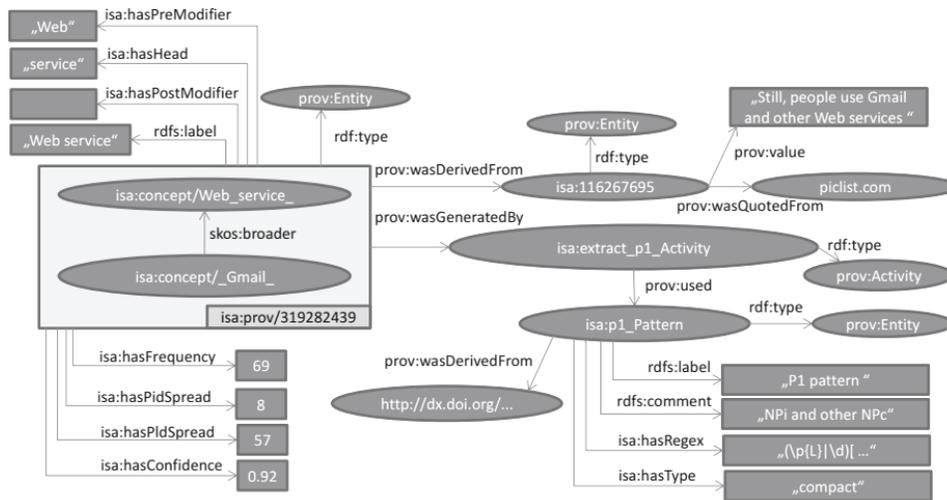


Figure 2.4: WebIsALOD Hypernymy Relation Example [76, p. 114]

Each hypernymy relation is stored in its own named graph which is indicated by the rectangular box.

2.3 The Ontology Matching Problem

This section covers the very core problem of this thesis: Ontology matching. First, general concepts are introduced. Afterwards, different levels of ontology heterogeneity are analyzed. In order to bridge to the world of data integration, schema matching and how it relates to ontology matching is explained subsequently. Thereafter, different techniques to ontology matching are presented and it is shown where

²⁰Endpoint: <http://webisxl.webdatacommons.org/sparql>

²¹The Internationalized Resource Identifier (IRI) extends the URI syntax with a richer character set (Unicode rather than ASCII) [46, p. 3]. URIs can be mapped to IRIs and vice versa [46, pp. 10-17].

the matcher of this thesis fits in. Lastly, areas of ontology evaluation and challenges within the process are covered.

2.3.1 General Concepts

Ontology The concept of ontologies has been introduced in 2.2.1. In the following, ontologies refer to their meaning in the context of the Semantic Web.

Correspondence A correspondence is a relation that holds between entities e_1 and e_2 which are from different ontologies. An entity can be a class or a property of an ontology. [49, p. 39] In its minimal form, a correspondence is a triple of the form (e_1, e_2, r) where r is the relation which holds between the entities. The relation is a set-theoretic one like *equivalence* ($=$), *disjointness* (\perp) or *less general* (\leq). Additionally, a matcher might assign an identifier (id) and a confidence value to a triple. [49, p. 43] In this thesis, the focus is on correspondences with equivalence relations.

Correspondences can be of different complexity: In its simplest form, a correspondence consists of the triple notation explained above, for instance: $(\text{onto1:Author}, \text{onto2:Writer}, =)$. Such simple relations can be insufficient and not expressive enough as there might be additional conditions such as restrictions or conversions. An example for three complex correspondences is given in figure 2.5 together with their translation in first-order logic in listing 2.2.

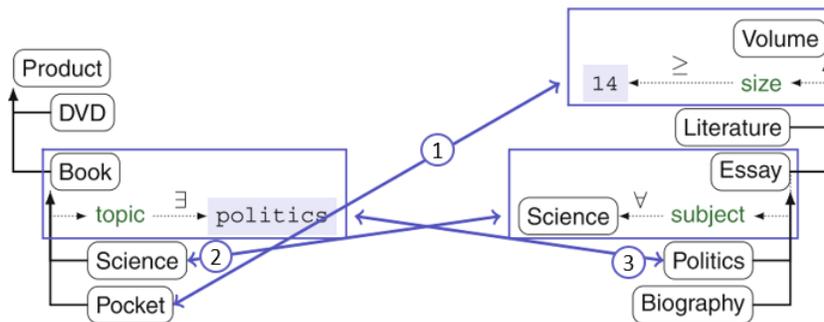


Figure 2.5: Complex Correspondences Example

This figure is taken from [49, p. 323] and adapted. First-order logic translations for the numbered correspondences can be found in listing 2.2.

- 1 $\forall x, \text{Pocket}(x) \equiv \text{Volume}(x) \wedge \text{size}(x, y) \wedge y \leq 14$
- 2 $\forall x, \text{Science}(x) \equiv \text{Essay}(x) \wedge (\forall y, \text{subject}(x, y) \Rightarrow \text{Science}(y))$
- 3 $\forall x, \text{Book}(x) \wedge \text{topic}(x, \text{politics}) \equiv \text{Politics}(x)$

Listing 2.2: Complex Correspondences in First Order Logic

The translations are given for the example in figure 2.5.

Those complex mappings require an elaborate format. Examples for such a format would be the *Semantic Web Rule Language (SWRL)* [81] or the *Expressive and Declarative Ontology Alignment Language (EDOAL)* which was originally known as *SEKT Mapping Language* [37] and *OMWG Ontology Mapping Language* [50]. [49, pp. 321-323; 327-333] This thesis concentrates on non-complex correspondences for which the alignment format of the Alignment API is used which is further explained in subsection 3.1.2.

Alignment The set of correspondences between ontologies is called *alignment*. An alignment is not restricted to a one-to-one (1:1) cardinality but can instead be of different cardinalities: One-to-one (1:1), one-to-many (1:m), many-to-one (m:1), or many-to-many (n:m) [158, p. 3]. Those are explained in more detail in the next paragraph. The goal of ontology alignment is, ultimately, to automatically obtain correct alignments between any given ontologies [49, pp. 39, 41].

Matching Restrictions The matching process can be subject to restrictions. There are multiple possible *arity restrictions* when ontology A is matched to ontology B :

1. One-to-One (1:1)
This restriction specifies that one element $e_1 \in A$ is matched to zero or one element $e_2 \in B$. Each element $e_2 \in B$ is matched to zero or one element $e_1 \in A$. When there are multiple options for correspondences and each correspondence has a confidence score, this problem is equivalent to the *maximum weighted bipartite graph matching problem* in mathematics.
2. One-to-Many (1:m) / Many-to-One (m:1)
This restriction specifies that one element $e_1 \in A$ is matched to zero or more elements $e_j \in B$. Each element $e_j \in B$ can, therefore, be matched to zero or more elements $e_1 \in A$.
3. Many-to-Many (n:m)
This restriction specifies that each element $e_i \in A$ is matched to zero or more elements $e_j \in B$.

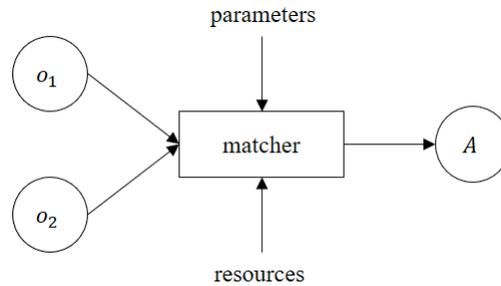


Figure 2.6: Matching Process According to Euzenat and Shvaiko [49, p. 41].

From an implementation viewpoint, there is not one exclusive option but multiple ways in implementing arity restrictions. [30, pp. 157-160]

Another alignment restriction is concerned about what can be matched: In a *homogeneous alignment* only resources of the same type are matched, for example ontology classes can only be matched to other classes but not to data or object properties. In *heterogeneous alignments*, on the other hand, any resource type can be matched to any other resource type. [57, pp. 235-237] In this thesis, the focus is on homogeneous alignments.²²

Ontology Matching The goal of the *ontology matching process* is to obtain an alignment A for a pair of ontologies o_1 and o_2 . This process is also known as *ontology alignment* or *ontology mediation* [36, p. 95]. This is achieved through a *matcher* which may use resources r (such as thesauri²³ or common knowledge) and which can be configured by setting parameters p (such as weights or thresholds). The matcher can be viewed as a function $f(o_1, o_2, p, r) = A$. This *matching process* is also depicted in figure 2.6.²⁴ [49, p. 41] [156, p. 1165]

It is also possible to apply a filter operation to the matcher output: Matchers can assign a confidence value to each correspondence which is usually in the $[0, 1]$ range. A threshold $t \in [0, 1]$ can then be defined to only add correspondences with a confidence $\geq t$ to the final alignment. [1, pp. 3-4]

²²This is due to the data sets used for evaluating the ontology matcher. The approach presented in this thesis is not restricted to homogeneous alignments and can also handle heterogeneous alignments.

²³A thesaurus groups lexemes by meaning. As opposed to a dictionary where the user tries to find the meaning or use of a lexeme, a thesaurus is used to find lexemes for a certain meaning. [34, p. 158] A well-known English thesaurus is *WordNet* [119]; an example for a German thesaurus would be *GermaNet* [63, 69].

²⁴Euzenat and Shvaiko include in their formal definition also an input alignment A' . As techniques utilizing A' are not discussed in this thesis, a slightly simplified version is presented here.

Concerning the methodology of ontology matching, no distinct superior methodology has emerged over the years – not even in the older field of data model schema matching [49, p. 55].

2.3.2 Ontology Heterogeneity

Differences in ontologies require a reconciliation process if interoperability is a desired property. The differences can occur at several levels. One important distinction are differences in the structure (syntax) and differences in the semantics. This observation is older than the Semantic Web itself and has already been made in the area of multidatabase systems [155, 97].

Several classification approaches exist to bring these observations into the broader context of ontologies, for example by Klein [99] or Hameed et al. [62]. Euzenat and Shvaiko consolidate different views on heterogeneity into four main types following Bouquet et al. [18]²⁵. Figure 2.7 displays a general overview of the different types of heterogeneity.

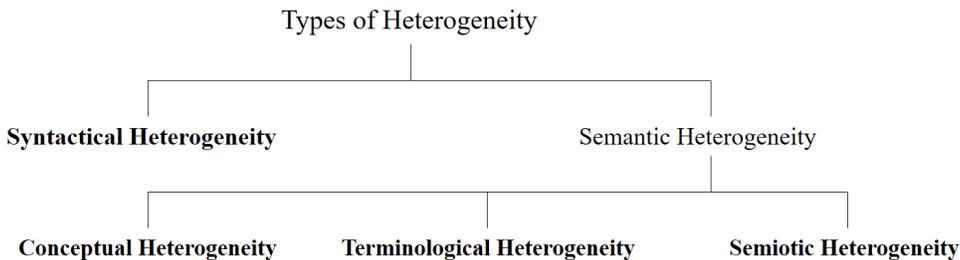


Figure 2.7: Ontology Heterogeneity

The grouping according to syntax and semantics is taken from Klein [99, p. 58], bold printed types are from Euzenat and Shvaiko [49, pp. 37 - 39]

Syntactic Heterogeneity Syntactic heterogeneity is used to refer to the difference in formalization of ontologies, i.e., when different ontology languages are used. In such cases, a transformation is required if interoperability is desired. [49, pp. 37-39]

Terminological Heterogeneity Terminological heterogeneity encompasses the situation where two identical concepts are described in distinct ontologies with different terms. This may be due to synonyms (*business partner* and *customer*) or due

²⁵Note that Euzenat is also co-author of this paper.

to different languages (*Finanzinstrument* and *Financial Instrument*), for instance. [49, pp. 37-39]

Conceptual Heterogeneity Conceptual heterogeneity in ontologies is due to differences in modeling. Concrete reasons are:

1. *Coverage*, originally called *partiality* [13, pp. 9-10], i.e., two ontologies describe different domains with the same level of detail. There may be an overlap between the two domains. [49, p. 38]
2. *Granularity*, originally called *approximation* [13, pp. 10-11], i.e., two ontologies describe the same domain but at different levels of granularity [49, p. 38].
3. *Perspective*, i.e., two domains describe the same domain but take different unique perspectives [49, p. 38] [13, pp. 11-13].

Semiotic Heterogeneity Semiotic heterogeneity characterizes the situation where concepts are described identically but are interpreted differently by users. This is due to the fact that interpretations may differ depending on the context in which they are made. [49, pp. 37-39]

2.3.3 Schema Matching

Doan et al. refer to *schema matching* and *schema mapping* as synonymous [44, p. 121]. A *semantic mapping* "relates a schema S with a schema T" [44, p. 122] and "[a] *semantic match* relates a set of elements in schema S to a set of elements in schema T" [44, p. 123].

2.3.4 Data Model Schema Matching and Ontology Matching

Even though there are differences in data modelling and ontology engineering (Spyns et al. mainly mention higher expressiveness, higher abstraction, and higher application independence of pure ontology models as opposed to database schemas [163, pp. 3-4]), there are also commonalities: According to the definitions of an ontology provided above, a conceptual data model and even a database schema can be regarded as an ontology. Techniques presented for ontology matching in this

thesis (and very often also elsewhere²⁶) can also be applied to schema matching of data models or databases. Straightforward approaches exist which allow to convert a database schema or entity-relationship diagram (ER diagram) into an ontology using OWL by applying a set of rules for example as outlined by Fahad [51]. Such a transformation is also performed within the scope of this thesis which is further described in subsection 4.6.1.

2.3.5 Techniques to Ontology Matching

There is not one superior matching technique or approach in matching ontologies.²⁷ Rather, there are different types and families of algorithms and approaches used. In this subsection, a categorization of techniques will be presented to better outline differences and similarities of algorithms and also to classify the matcher developed in this thesis. In 2005, Shvaiko and Euzenat presented two classifications for matching approaches [157] which were revised in 2013 [49].

The first classification approach, called *Granularity/Input Interpretation*, differentiates matchers according to the granularity, which can be either element-level (analyze entities/instances in isolation) or structure-level (analyze the ontology structure), and then according to whether syntactic or semantic techniques are used (*Input Interpretation*). Syntactic techniques use a structured algorithm whereas semantic techniques apply formal semantics (see section 2.1).

The second classification approach, called *Origin/Kind of Input*, first differentiates according to whether context (i.e., external resources) or content (i.e., internal resources like the structure or instances) is used (*Origin*) and then further distinguishes different characteristics of the origin (*Kind of Input*). [49, pp. 74-82] Both classification approaches are depicted in figure 2.8.

²⁶Euzenat and Shvaiko already write in the preface of their book *Ontology Matching* that "though we use the word ontology, the work and the techniques considered in this book can equally be applied to database schema matching [...] and other related problems" [49, p. viii]. Similarly, in Hepp et al.'s textbook *Ontology Management*, Euzenat, Mocan, and Scharffe write: "When we talk about ontologies, we include database schemas and other extensional descriptions of data [...]" [70, p. 178]. There is also literature where ontology matching is viewed as a form of schema matching, for example in *Schema Matching and Mapping* by Bellahse, Bonifati, and Rahm [12]. In the latter book, schemas and ontologies are both viewed as metadata models between which mappings can exist [12, p. v].

²⁷This can easily be seen when looking at the different algorithms applied at campaigns by the Ontology Alignment Evaluation Initiative.

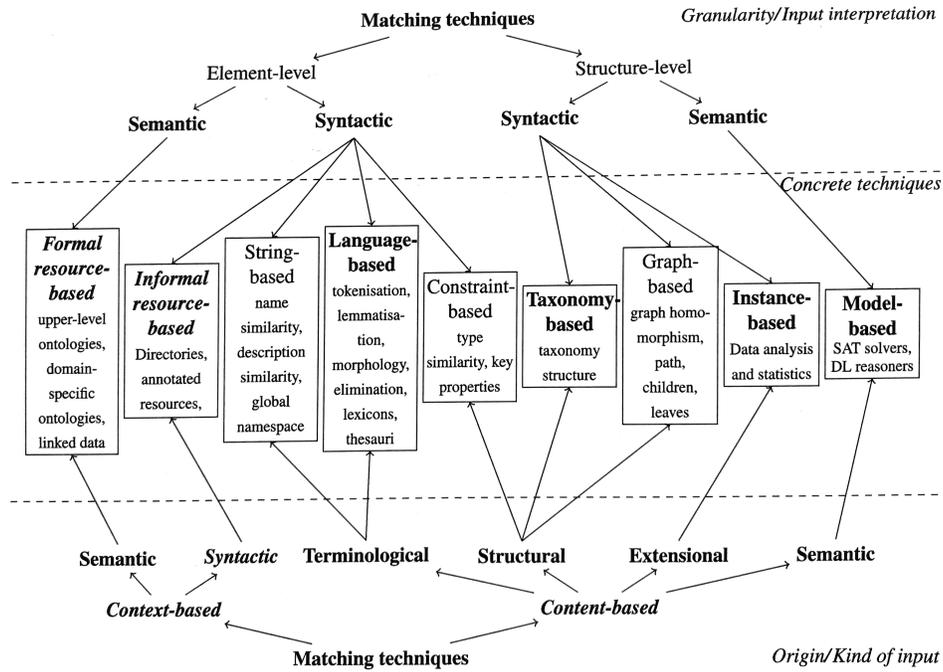


Figure 2.8: Ontology Matching Classification Approaches [49, p. 77]
 Bold printed concepts are newly introduced compared to [136] and italic-bold printed concepts were added in the 2013 version [49]. Note that the original figure [157, p. 155] did not have formal and informal resource-based techniques, those were introduced in the newest version.

Formal Resource-Based Techniques Formal resource-based techniques make use of external ontologies (which can also be domain specific). It is also possible to use linked data. [49, p. 80]

Informal Resource-Based Techniques Informal resource-based techniques utilize informal resources such as pictures or encyclopedia pages. Ontology entities can be related to such resources. [49, pp. 80-81]

String-Based Techniques A very old class of techniques are string-based techniques which use annotations – such as names, labels, and descriptions – to calculate similarities between resources. The underlying intuition is that similar words are used to describe similar concepts. [49, p. 79]

Language-Based Techniques²⁸ String-based techniques presented above do not require that the language is known in order to be applied. Language-based techniques, on the other hand, consider text encoded in specified language. Linguistic techniques, like lemmatization or tokenization, can be used here, for example. Phonetic methods, such as *Soundex* [80] or *Kölner Phonetik* [132], also fall into this category. It is, furthermore, possible to exploit external, language-based resources, such as thesauri or lexicons. [49, p. 80]

Constraint-Based Techniques Constraint-based techniques check internal constraints which apply for entities such as cardinality or data types [49, p. 80].

Taxonomy-Based Techniques Taxonomy-based techniques apply graph algorithms on the inheritance structure of the resources. The underlying intuition is that concepts that are connected by inheritance are similar. [49, p. 81]

Graph-Based Techniques Graph-based techniques also view the ontology as a graph. Compared to taxonomy-based techniques, they consider all kind of information within the graph. Pattern matching methods count as graph-based techniques, for instance. [49, p. 81]

Instance-Based Techniques Depending on the use case, instances of the ontologies to be matched might be available. Comparing concrete instances can help to calculate distances of resources in the ontologies. Such approaches are referred to as *instance-based techniques*. [49, p. 82]

Model-Based Techniques Lastly, model-based techniques exploit reasoning and propositional satisfiability in order to match two ontologies. [49, pp. 81-82]

This thesis explores approaches based on linked data (see 2.2.2) which is an element-level approach utilizing semantics.

Rather than content features, context is used by applying semantics. The context that is used is available in a formalized format: The knowledge is represented using a vocabulary that has an underlying ontology as opposed to free text of an article for instance.

According to the two classification approaches presented above, this approach falls in the *formal resource-based* class of techniques (see 2.8).

²⁸In an earlier version [157, p. 155], external linguistic resources were explicitly differentiated from plain language-based techniques. The latest version [49, p. 77] counts everything concerned with the actual language into this category and does not explicitly make this differentiation.

2.3.6 Evaluation of Ontology Alignments

Measures Ontology alignments are commonly evaluated on the basis of *reference alignments*, i.e., an annotated gold standard of correspondences. Typical machine learning and information retrieval evaluation measures, like *Precision*, *Recall*, and *F-Measure*, are used to judge the quality of the alignment; formulas and further details about those can be found in appendix C.1.

When evaluating multiple data sets D at once, there are two options for giving one overall performance number: Macro average and micro average. Macro average simply averages scores regardless of the individual data sets' size. The formula is given in an exemplary way for F_1 in equation 2.2:

$$\frac{\sum_{d=1}^{|D|} F_1(d)}{|D|} \quad (2.2)$$

where $F_1(d)$ is the obtained F_1 score on data set $d \in D$ and $|D|$ is the total number of data sets.

In order to calculate the micro average, one contingency table is built for all data sets by adding all true positives, true negatives, false positives, and false negatives of each individual data set. Then, precision, recall, and F_1 can be calculated by using this table (see table C.1 in the appendix). [120, p. 185]

OAEI In order to compare various matchers in a fair setting, common reference alignments are required. The Ontology Alignment Evaluation Initiative (OAEI)²⁹ tackles this problem by providing several reference alignments and carrying out campaigns every year since 2004. Participants can evaluate their matchers in several tracks. [49, p. 288] One major goal of the OAEI is to create transparency and "to allow anyone to draw conclusions about the best matching strategies" [156, p. 1170].

For the alignment evaluation, the *Semantic Evaluation At Large Scale (SEALS)*³⁰ platform is used. Starting in 2017, the OAEI is beginning to use the *Holistic Benchmarking of Big Linked Data (HOBBIT)*³¹ platform where users eventually will be able to upload and evaluate matching systems [138, pp. 5-10].

2.3.7 Challenges of Ontology and Schema Matching

Different people describe concepts differently and create different schemas [44, pp. 124-125]. This leads to *semantic heterogeneity* on various levels as outlined

²⁹see <http://oaei.ontologymatching.org>

³⁰see <http://seals-project.eu/>

³¹see <https://project-hobbit.eu/>

in subsection 2.3.2. The resolution of the semantics itself is a challenge, as the semantics within a schema might not be fully captured by names or definitions. Additionally, schema matching is a subjective task where different people might have different opinions concerning a particular correspondence. [44, pp. 125-126] In 2005, Shvaiko and Euzenat formulated 10 challenges for ontology matching [156], among which discovering missing background knowledge was nominated [156, pp. 1171-1172]. Eight years later, the authors reaffirm eight of the ten challenges [158]³²; discovering missing background knowledge is still an open issue [158, pp. 170-171]. Among current challenges are also non-functional ones, like the performance³³ of ontology-matching techniques, or alignment infrastructure and support [158, pp. 169, 174].

Handling syntactical heterogeneity is rather a minor problem compared to resolving semantics.

2.4 Natural Language Processing

In this section, aspects of natural language processing are introduced that are relevant for the present paper.³⁴ As the implementation of this thesis builds upon a specific technology, namely *word2vec*, the latter approach is explained in more detail in subsection 2.4.2.

2.4.1 General Concepts

Natural Language Processing Natural language processing (NLP) "is the attempt to extract a [...] meaning representation from free text" [94, p. 1]. It is, hence, about capturing semantic content from text.

The Distributional Hypothesis The dictum "you shall know a word by the company it keeps", which is attributed to linguist J. R. Firth, stresses the role of the

³²In appendix C.2, all dropped and open challenges are listed.

³³The currency of this topic can immediately be seen when reviewing the OAEI 2017 LargeBio track where not even half of the matchers were capable of matching the ontologies within a timeframe of less than four hours; see <http://www.cs.ox.ac.uk/isg/projects/SEALS/oeai/2017/results/>.

³⁴Due to the scope of this thesis and the extent of the field of natural language processing, the introduction is heavily reduced to the parts relevant for the course of this work. A good and comprehensive introduction can be found in *Natural Language Processing and Text Mining* by Kao and Poteet (eds) [93].

context of a word when analyzing its meaning [34, p. 160]. The distributional hypothesis builds on that observation and states that a word is similar to those words with which it frequently co-occurs [65, p. 156]. Distributional Semantic Models (DSMs) make use of that finding by approximating words with co-occurrence patterns [23, p. 2]. Despite the wide application of the distributional hypothesis, there is also criticism. One point of critique is that concepts are actually more complex than the simple co-occurrence of words.³⁵ [22, p. 150]

Bag of Words A text or a document can be viewed as a collection of its containing words which is known as the *bag of words model*. Information about the position of words is lost but information about the multiplicity of the words, i.e., their counts, is retained. [113, p. 117] From the bag of word model a simple boolean feature vector can be derived.

Stopwords A stopword is a word that appears usually in a high frequency in texts and is, therefore, not valuable for many information retrieval tasks. An example would be the article *the* which appears in almost all English texts; its occurrence in a document says close to nothing about the content. In many systems those words are not indexed at all. For a practical use, such words are often collated in so called *stoplists* which are utilized in applications. Stoplists can be generic or specific to a dedicated domain. [137, pp. 2794-2795] [113, pp. 26-27]

Distances in Vector Space When documents are transferred into vector space, their similarity is usually calculated using *cosine similarity*. One advantage over the Euclidean distance is that the length of the document does not influence the outcome. Given two documents d_1 and d_2 and their vector representations $\vec{V}(d_1)$ and $\vec{V}(d_2)$, the cosine similarity is given as follows [113, p. 121]:

$$\text{sim}(d_1, d_2) = \frac{\vec{V}d_1 * \vec{V}d_2}{|\vec{V}d_1||\vec{V}d_2|} \quad (2.3)$$

In this application the elements are never negative which implies $\text{sim}(d_1, d_2) \in [0, 1]$ where a value of 1 indicates maximum similarity.

³⁵Stevan Harnad draws attention to the aspect that symbols need to get their meaning from *some-where* which is not the symbol system itself because this would be like "trying to learn Chinese from a Chinese/Chinese dictionary alone" [64, p. 335]. This problem is referred to as "Symbol Grounding Problem" [64]. There are approaches that try to find a solution to this problem e.g. by learning embeddings that combine vision and text such as [159].

Embeddings Boolean feature vectors have severe disadvantages such as data sparsity and high dimensionality. One solution is to derive a vector in a lower dimensional space with the goal to keep as much information as possible. A very straight-forward approach is to perform a matrix decomposition, e.g. by applying *Singular Value Decomposition (SVD)*. [113, pp. 403-408] Similarities between the vectors can be calculated by using cosine similarity (see equation 2.3).

2.4.2 Neural Language Models: Word2Vec

One of the most well-known neural language models is the *word2vec model* [117, 118]. Two approaches are presented in order to calculate the embeddings: The *Continuous Bag-of-Words Model (CBOW)* and the *Continuous Skip-gram Model (SG)*.

Continuous Bag-of-Words Model Given the context k of a word w , the continuous bag-of-words model is trained to predict w where k are preceding and succeeding words $k = w_1, w_2, w_3, \dots, w_c$ and where c is the size of the training context. Here, c is also referred to as *window*. The overall architecture is depicted in figure 2.9. The objective of CBOW is to maximize the average log probability [117, p. 4] [143, p. 117]:

$$\frac{1}{T} \sum_{t=1}^T \log p(w_t | w_{t-c} \dots w_{t+c}) \quad (2.4)$$

where p is obtained by applying the softmax function:

$$p(w_t | w_{t-c} \dots w_{t+c}) = \frac{\exp(\bar{v}^T v_{w_t})}{\sum_{w=1}^V \exp(\bar{v}^T v_w)} \quad (2.5)$$

where v_w is the output vector of word w , V is the vocabulary of words, and \bar{v} is the averaged input vector of all the context words:

$$\bar{v} = \frac{1}{2c} \sum_{-c \leq j \leq c, j \neq 0} v_{w_{t+j}} \quad (2.6)$$

Skip-gram Model As opposed to CBOW, the skip-gram model predicts the context k given the target word w . The overall architecture is depicted in figure 2.9. The objective here is to maximize the average log probability [118, pp. 2-3]:

$$\frac{1}{T} \sum_{t=1}^T \sum_{-c \leq j \leq c, j \neq 0} \log p(w_{t+j} | w_t) \quad (2.7)$$

where p is obtained by applying the softmax function:

$$p(w_{t+c}|w_t) = \frac{\exp(v_{w_o}^T v_{w_I})}{\sum_{w=1}^W \exp(v_w^T v_{w_I})} \quad (2.8)$$

where v_w and v_w are the input and output vector representations of w and W is the number of words in the vocabulary.

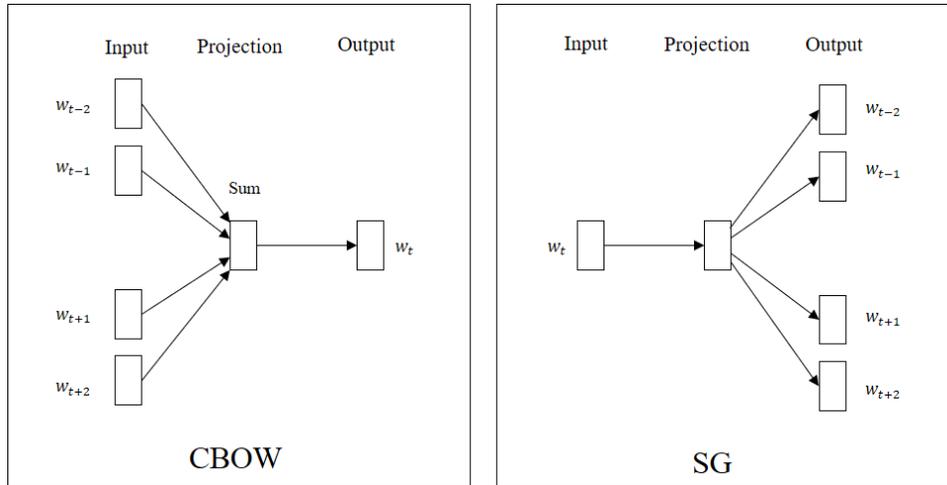


Figure 2.9: Word2Vec Architecture [117, p. 5]

On the left side the Continuous Bag-of-Word Model (CBOW) is depicted, on the right side the Skip-gram Model (SG); the figure is adapted.

2.5 Related Work

In this section, related work is presented with a focus on two research areas: (1) Propositionalization techniques and (2) ontology matching techniques that use external resources. *node2Vec* and *RDF2Vec* are explained in more detail in research area (1) because a similar approach is applied in this thesis.

2.5.1 Propositionalization in the Context of Knowledge Graphs

Data mining algorithms usually require data to be translated to a propositional *feature vector* $\langle f_1, f_2, \dots, f_n \rangle$ where features are either binary ($f_i \in \{true, false\}$), nominal ($f_i \in S$ where S is a finite set of symbols) or numerical ($f_i \in R$) [143, p. 68]. An RDF graph like that of the WebIsA data set can, hence, not be used right away in a data mining algorithm but instead, the information contained in the

graph is to be translated into feature vectors. This process which translates data into a feature-based format is known as *propositionalization* [101, p. 262].

Traditional propositionalization approaches for Linked Open Data are supervised which means that automatic feature generation is not possible but SPARQL queries have to be formulated by the user [145, p. 7]. The *Linked Data Data Miner (LiDDM)* application [121], for instance, offers a UI which facilitates the process of using linked data in data mining tasks but still requires the user to build the queries herself. Similarly, the *semweb* plugin allows to load and process data in RapidMiner³⁶, a click-and-run data mining tool, after having formulated a SPARQL query [95].

In terms of unsupervised approaches, a simple approach is to apply binary features indicating whether a specific relation or value of a property is existing [145, pp. 7-9].

Another, more advanced, option are kernel functions. They form a class of machine learning algorithms which are able to quantify the similarity of graphs by typically relying on the number of patterns the graphs have in common [60, pp. 467-468]. An example for a graph kernel that can be applied to RDF data would be the *Weisfeiler-Lehman Graph Kernel* [154]. The *RapidMiner Linked Open Data Extension* [144] is capable of calculating features in an unsupervised fashion and is also capable of applying kernel functions.

A different class of approaches are translation-based approaches. Those generally assume a multi-relational data set consisting of a set of entities E and a set of edges L as well as triples in the form $(head, label, tail)$ which is usually stated as (h, l, t) where $h, t \in E$ and $l \in L$. The most well-known approach is *TransE* [17]. The general idea is that "if (h, l, t) holds, then the embedding of the tail entity t should be close to the embedding of the head entity h plus some vector that depends on the relationship l " [17, p. 2788]. The embeddings for h, l and t are all in the same dimension. The general objective for learning the embeddings is that $h + l = t$ given that (h, l, t) holds [17, p. 2790]. The model has been further developed and inspired similar approaches. Wang et al. note that *TransE* does not handle reflexive, one-to-many, many-to-one and many-to-many properties well and propose another approach named *TransH* [177]. Many more approaches based on *TransE* have been developed such as *TransA* [86], *TransF* [42], or *TransD* [85].

Recently, propositionalization approaches have been presented that also exploit advances in natural language processing, namely *word2vec*. Notable members of this group are *node2vec* [59] and *RDF2Vec* [146] which are explained in more detail in the following two paragraphs.

It is important to note that the choice of the propositionalization strategy can have

³⁶see <https://rapidminer.com/>

a major impact on the results [145, p. 14].

Node2Vec Grover and Leskovec present an approach that allows to derive embeddings for nodes in a graph using the skip-gram model. Given $G = (V, E)$ where V is a set of vertices and E is a set of edges, a sequence of nodes that are connected by edges, i.e. a *walk* [24, p. 57], is interpreted as sentence that is used as input for the skip-gram model. The approach presents a semi-supervised, biased walk generation strategy that requires two parameters to be set in advance: (1) A return parameter p is used to control the likelihood of a walk revisiting the previous node. This is particularly important for undirected graphs. (2) A parameter q is used to control how fast the walk spreads. This is done by rewarding or punishing walks that have a large shortest distance between the starting and the end node. [59, pp. 858-859] The obtained sentences are used to train the embeddings. The implementation is publicly available on GitHub.³⁷

RDF2Vec Ristoski and Paulheim present an approach which also adapts neural language models for RDF data to derive graph embeddings, called *RDF2Vec* [146]. Compared to node2vec, RDF2Vec is explicitly focused on RDF data and also considers edges. Given an RDF graph $G = (V, E)$ where V is a set of vertices and E is a set of directed edges, a set of walks is generated for each vertex $v \in V$ of depth d , where d refers to the number of edges in the walk which is also known as *length* [24, p. 57]. In RDF2Vec, two possible approaches are presented to generate the graph walks: (1) A breadth-first algorithm used iteratively to generate random walks or (2) an adaption of the Weisfeiler-Lehmann algorithm [38, 39]. In comparison to node2vec, the edges are labelled and appear in the sentences. The obtained sentences are then used to train word2vec embeddings. The implementation is publicly available.³⁸ In [147], the embeddings are also applied to document modeling and recommender systems. An extended approach of RDF2Vec are *Biased Graph Walks for RDF Graph Embeddings* [32]. The approach is generally the same as that described in the original paper in the sense that it is using random walks – but with the difference that edges receive weights with the goal of capturing the most important information. Different strategies for weighting are evaluated relying on frequencies (e.g. of predicates) and also on different adaptations of the PageRank algorithm [127]. Despite the advanced weighting techniques used, the authors admit that "[u]nexpectedly, the *Uniform Weight* strategy also yields competitive results" [32, p. 8].

³⁷see <https://github.com/aditya-grover/node2vec>

³⁸see <http://data.dws.informatik.uni-mannheim.de/rdf2vec/>

2.5.2 Ontology Matching Utilizing External Resources

While some matching methods exclusively use information contained within the ontologies such as names or the structure, others exploit externally available knowledge. A very common source of external knowledge are thesauri that contain semantic relations such as WordNet [119]. WordNet is a database of English words grouped in sets which represent one particular meaning, called *synsets*. The database is publicly available³⁹ and there is an online tool for querying the synsets⁴⁰. The thesaurus is heavily used for ontology matching.⁴¹ Current matching solutions, like those participating in the OAEI 2017, are also using this particular resource, for example *LogMap* [87] or *KEPLER* [92].

LogMap is noteworthy, as the matcher regularly participates in the OAEI campaigns, often in different flavors like *LogMapBio* or *LogMapLite* [90, pp. 153-154]. The matching algorithm starts by performing a lexical indexation. During this step, external resources such as WordNet are used. Then, the matcher performs a structural indexation and computes initial anchor mappings which are considered to be high-confidence mappings. A repair process follows where reasoning is used to detect and repair unsatisfiable classes. The matcher extends the contexts for matching by applying a commonly used string distance metric named *String Metric for Ontology Alignment (SMOA)* [166]. Due to the applied heuristics (starting with anchor mappings and extending those), the matcher can also handle very large ontologies. [87, pp. 275-276]

Despite the fact that WordNet is widely used, the biggest weakness of the thesaurus is coverage [180, p. 34]: For entities which are not found in WordNet, a semantic similarity score cannot be calculated.

Another source of external knowledge are upper ontologies. In 2010, Mascardi et al. describe an approach to ontology alignment utilizing upper ontologies: Entities are mapped to the upper ontology and similarity is calculated within the upper ontology; the upper ontology is used as "semantic bridge" [114, pp. 609, 612-613]. Only very few matchers make use of publicly available knowledge on the Web. *BLOOMS* [84] was the first approach which exploited Wikipedia as upper ontology. The approach utilizes the Wikipedia category hierarchy to build trees which are then compared by using an overlap calculation. [84, pp. 404-406]

Also for multilingual ontology matching external resources from the Web can be

³⁹see <https://wordnet.princeton.edu/download>

⁴⁰see <http://wordnetweb.princeton.edu/perl/webwn>

⁴¹To get an impression, one can review pages 271-277 in the book *Ontology Matching* [49] where 87 matching approaches are compared in table 8.1. WordNet is explicitly cited as a resource for 26 matchers.

exploited. Lin and Krizhanovsky [111] parse *Wiktionary*⁴², a free online dictionary by the Wikimedia Foundation, and provide a SPARQL endpoint for querying the translations. They integrate the translation process into the the *Context-base Ontology Matching System (COMS)* [110], a matcher that combines a string matching strategy, a lexical matching strategy, and a structural matching strategy [110, pp. 1102-1105]. As a consequence, the matcher can perform multi-lingual ontology alignment.

Other common external Web sources for translations are *Microsoft Bing* which is used by KEPLER [92] or *Google Translator* which is used by LogMap for multi-lingual ontology matching for instance.

WeSeE-Match [128] uses actual and current Web data by querying the Microsoft Bing Search API in order to build a describing document consisting of the retrieved website titles and excerpts for each element. An element-level comparison is performed by computing TF-IDF similarity scores. It could be shown that the approach is very sensitive to the underlying search API that is used: In the OAEI 2013 campaign, the approach dropped Bing for a non-commercial service resulting in a drastic decline of performance [129].

WikiMatch [73] is a matcher utilizing Wikipedia as external source by querying the Wikipedia search API for concepts and by calculating the overlap of the returned articles. As inter-language links connect articles in different languages, the matcher is also capable of matching ontologies in different languages. The approach was evaluated in the OAEI 2012 [75], 2013 [58], and also recently in a third version in 2017 [72].

Zhang et al. [180] present a matcher which also uses Wikipedia but in a different way: They train word embeddings with Wikipedia as corpus and calculate semantic similarity on element level.⁴³ The best closest correspondences are added to the resulting alignment. The approach is evaluated on the OAEI 2013 Benchmark and Conference data set but the authors did not participate in the actual campaign.⁴⁴ They conclude that their approach is better than WordNet based methods and that word embeddings are good in dealing with synonyms [180, p. 41].

Similar to the latter approach, Swoboda et al. use word2vec for the combination of domain-specific taxonomies which is similar to ontology alignment. They conclude that their approach can "alternatively be used for [...] automated alignment of ontologies and semantic integration" [167, p. 134].

⁴²see <https://www.wiktionary.org/>

⁴³Unfortunately, information about the training process used is not given. Instead, the authors write that they do not intend "to describe the training [in] detail because it is complicated and not closely related to [the] ontology matching task itself" [180, p. 37].

⁴⁴Their approach is not listed on the OAEI Web page and does also not appear in the official publications for 2013 [58] and 2014 [45].

Chapter 3

Implementation

3.1 Matcher Architecture

3.1.1 Overview

In this section, the structural overview of the implemented matcher is discussed. Figure 3.1 depicts the matching process developed in this thesis. The implementation accepts two ontologies (O_1 and O_2) which are first read. In a second step, textual descriptions (*annotations*) for each resource are extracted from the ontologies. A resource can be one of the following: A class, a datatype property, or an object property. To avoid heterogeneous mappings (such as a class mapped to a property), the implemented matcher treats each resource type separately. As one resource may have multiple annotations, a strategy is required to compare resources. The chosen approach creates a Cartesian product between two sets, applies a comparison metric, and selects the best score. This approach is described in subsection 3.6.3 in more detail.

For reasons of performance, a filter will sort out direct String matches which will not run through the subsequent steps but will rather be used later by the matching strategy. The remaining annotations will be linked to concepts on ALOD. How the linking process works is further explained in section 3.2. As one annotation may contain more than one concept, a strategy is required here as well in order to compare two annotations. The handling of this issue is addressed in subsection 3.6.2. A feature generator will then calculate a similarity score for the cross combination of the annotations for each resource in O_1 and O_2 whereupon the best value will be set as confidence. In the scope of this thesis, multiple features have been developed and evaluated. They are presented in more detail in section 3.3. The correspondence scores are eventually used by a matching strategy to determine which correspondence makes it into the final alignment (A). The implemented matching

strategies are explained in subsection 3.6.1. The APIs and frameworks used for the implementation are briefly described in the following subsection (3.1.2).

3.1.2 Used APIs and Frameworks

For the implementation of this thesis external APIs and frameworks are used. The most important ones are presented in the subsequent paragraphs.

Alignment API The general structure of the matcher follows the Alignment API [35]. The API is commonly used in the field of ontology matching and also recommended by the OAEI. It is implemented in Java and (despite its name) also contains sample implementations and logic parts. Following the API allows the matcher to be data set independent and, furthermore, to use OAEI data sets without data translations. The API defines an `AlignmentProcess` interface for all matchers with an `align` method. An `Evaluator` can evaluate an alignment process given a reference alignment. `AlignmentVisitors` can render an alignment.

The API also defines an XML format for alignments: A `Cell` represents a correspondence and contains at least `entity1`, `entity2`, and a `relation`. Optionally, a confidence (`measure`) can be assigned to a cell. An example for one correspondence can be found in listing 3.1.

```
1 <map>
2   <Cell>
3     <entity1 rdf:resource="http://example1.com/
4       article"/>
5     <entity2 rdf:resource="http://example2.com/
6       publication"/>
7     <measure rdf:datatype="xsd:float">0.5</measure>
8     <relation>=</relation>
9   </Cell>
10 </map>
```

Listing 3.1: Alignment Format Example as Defined by the Alignment API¹

Apache Jena Originally developed by HP Labs, *Jena* [29] is a commonly used Java framework for the Semantic Web. In 2010, the project was handed over to

¹The given example is only an excerpt of a possible alignment, the full alignment file also contains information about the type of the alignment and the ontologies. A good overview of the format can be found on the official page (see <http://alignapi.gforge.inria.fr/format.html>).

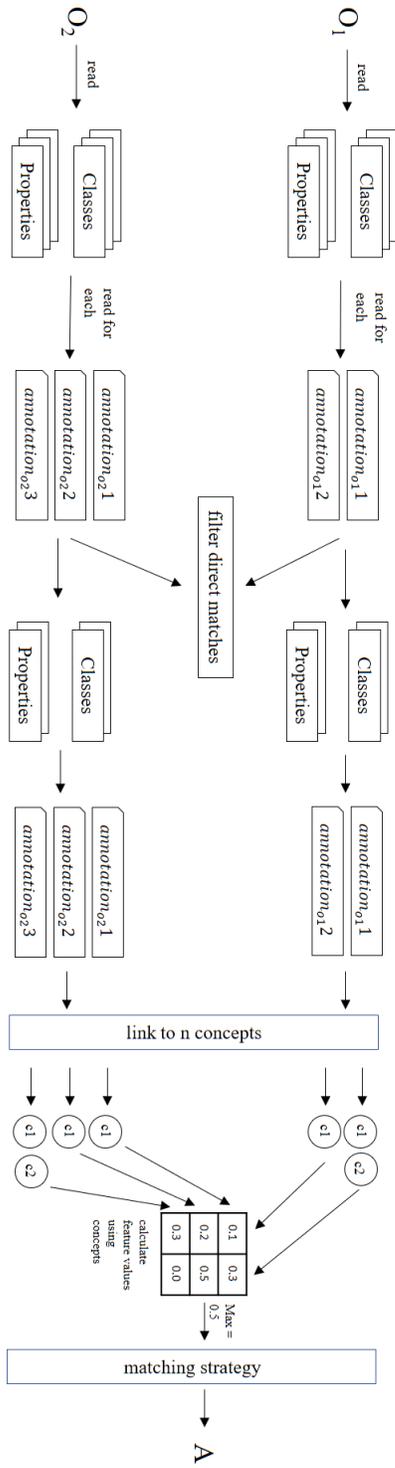


Figure 3.1: Matcher Structure Overview

the Apache Software Foundation [4]. It is available for free under a public license.² The framework allows (among other functions) to query RDF data³ and also provides a triple store, called *TDB*⁴. All SPARQL data operations concerning RDF data within the implementation of this thesis are achieved with the Jena framework.

Gensim Gensim [139] is a free Python library which allows to calculate embeddings efficiently.⁵ Among the supported algorithms, there is also a word2vec implementation.⁶ The framework is available for free under a public domain license.⁷ Gensim is used in this thesis to implement a modification of the RDF2Vec algorithm (see section 3.4).

SQLite SQLite⁸ is a lightweight transactional relational database management system (RDBMS) that does not require a server but accesses the required file directly on the client; the software is platform-independent and freely available under a public domain license [102] [3, pp. 1-9]. In the scope of this thesis, the framework is used to store vectors efficiently.

MapDB Similar to SQLite, MapDB⁹ allows to persist and retrieve data on disk. Unlike SQLite, the framework does not provide full RDBMS functionalities but rather fast access of disk-persisted data structures, such as HashMaps. The software is under a public domain license and open-source [100, p. 1]. For the present implementation, the framework is used to store vectors efficiently.¹⁰

3.2 Linking to LOD Resources

Unlike Wikipedia, the ALOD data set does not have a search feature which can be exploited. In contrast to WikiMatch or BLOOMS, which both use an externally provided search functionality [84, p. 406] [74, p. 3], a sophisticated linking mechanism for the ALOD data set has to be implemented for this particular use case.

²see <https://jena.apache.org/>

³see <https://jena.apache.org/documentation/query/index.html>

⁴see <https://jena.apache.org/documentation/tdb/index.html>

⁵see <https://radimrehurek.com/gensim/index.html>

⁶see <https://radimrehurek.com/gensim/models/word2vec.html>

⁷see <https://radimrehurek.com/gensim/about.html>

⁸see <http://sqlite.org/index.html>

⁹see <http://www.mapdb.org/>

¹⁰The framework was additionally introduced due to a library incompatibility of SQLite with the SEALS evaluation platform.

An ontology consists of classes or properties which (usually) have assigned labels. Given a label of a concept, an algorithm is required that is capable of linking the concept to a Web resource. This section focuses around the task of linking labels of local ontology resources to concepts on the Web.¹¹ Although presented for a particular data set in this section, the algorithm can also be employed to create links to resources in other data sets with few modifications.¹²

A naïve approach would be to use the label as it is to look up either labels or the Web resource directly. This leads to very few matches because of data set specific idiosyncrasies. In the ALOD Classic data set, for instance, all labels are in lower cases and resource fragments start and end with an underscore like `_piano_`¹³; when looking at composite resources, however, there is no leading underscore such as in `piano_player_`¹⁴. On DBpedia, on the other hand, labels and also resource fragments do contain upper case letters; there are no leading or trailing underscores as well.

In addition, ontology and data model labels often contain composite concepts. The OAEI Anatomy data set¹⁵, for example, contains the label *Hepatic Flexure of the Colon*. This label cannot be found in the ALOD data set in its full form. Yet, the data set does contain *Hepatic Flexure*¹⁶ and *Colon*¹⁷. A sophisticated algorithm is required here that recognizes partial concepts contained within a label. In the implementation developed in this thesis, this task is realized by a class implementing the `LabelToConceptLinker` interface. Because of data set specific idiosyncrasies, every label has to be cleaned of illegitimate characters; this is implemented as a data set specific cleaner which has to implement the interface `StringCleaner`.

Modification Sequences The `LabelToConceptLinker` gets a list of operations (`List<StringModifier>`) to try one after each other if the previous operation failed. One such modifier operation could be to remove all numbers, to remove stopwords, or to replace certain characters (like "-") with others (like "_"). When a concept was found, it is returned and the remaining operations are not ex-

¹¹For simplicity, the term *label* is used here to refer to a human-readable textual identifier of a resource that does not have to be represented through `rdfs:label`. In fact, the implementation uses all textual representations available and picks the most appropriate one. This process is described in detail later in subsection 3.6.3.

¹²A concrete example can be found in section 4.3 where the algorithm is used to create links not only to ALOD but also to DBpedia.

¹³see http://webisa.webdatacommons.org/concept/_piano_

¹⁴see http://webisa.webdatacommons.org/concept/piano_player_

¹⁵see <http://oaei.ontologymatching.org/2017/anatomy/index.html>

¹⁶see http://webisa.webdatacommons.org/concept/hepatic_flexure_

¹⁷see http://webisa.webdatacommons.org/concept/_colon_

ecuted anymore. The modification sequences for the ALOD data set can be found in appendix A.1.

Left-to-Right Tokenization If the label as a whole cannot be found on the data set but does contain multiple concepts, the label has to be split. Depending on the nature of the ontology, tokens are obtained differently. The simple term *car dealer*, for instance, might be encoded as `car dealer`, `car_dealer`, or `carDealer`. The implementation can automatically detect and handle underscores, camel case, and space separation. This is sufficient for most ontologies.¹⁸

The underlying assumption of the linking mechanism is that longer labels are more precise and carry more value. Hence, when a string operation on the whole label succeeds, no tokenization is performed. When the whole label cannot be found after execution of the modification sequence, however, it is assumed that multiple resources (sub-concepts) are contained within the label. Given the example of the label *United Nations Peacekeeping Mission in Mali*, a mechanism is required that finds the longest possible substrings (i.e., *united nations*¹⁹, *peacekeeping mission*²⁰, *mali*²¹) rather than searching for each token individually (i.e., *united*, *nations*, *peacekeeping* etc.).

As Western languages are read from left to right, the tokenizer will start chopping tokens from the right to get the longest possible token from the left. After each chopping, all string modifications are reapplied. If a lookup for a token was successful, the found tokens are removed and the process starts from the beginning until there are no tokens to chop anymore. An exemplary lookup process is depicted in table 3.2. Note that the sub-concept detection process is only triggered after the full term could not be linked.

As this process can be very expensive on labels with many tokens (a label consisting of 20 tokens can be cut up to $\binom{20}{2} = \frac{20!}{2!(20-2)!} = 190$ times), the maximal size of a lookup token can be set (e.g. if the maximal number is 2, this leads to only $\binom{10+1-2}{2} = \frac{9!}{2!(9-2)!} = 36$ potential cuts on the same label). Additionally, the label can be cut after a certain amount of characters.

¹⁸Tested were 9 different OAEI ontologies (see also section 4.3).

¹⁹see http://webisa.webdatacommons.org/concept/united_nations_

²⁰see http://webisa.webdatacommons.org/concept/peacekeeping_mission_

²¹see http://webisa.webdatacommons.org/concept/_mali_

Action	Current Lookup String	Status
String Modifier: LC	united nations peacekeeping mission in mali	false
String Modifier: LCSR (stopword detected)	united nations peacekeeping mission mali	false
Tokenizer: Cut Last Word String Modifier: LC	united nations peacekeeping mission in	false
String Modifier: LCSR (stopword detected)	united nations peacekeeping mission	false
Tokenizer: Cut Last Word String Modifier: LC	united nations peacekeeping mission	false
Tokenizer: Cut Last Word String Modifier: LC	united nations peacekeeping	false
Tokenizer: Cut Last Word String Modifier: LC	united nations	true
String Modifier: LC	peacekeeping mission in mali	false
String Modifier: LCSR (stopword detected)	peacekeeping mission mali	false
Tokenizer: Cut Last Word String Modifier: LC	peacekeeping mission in	false
String Modifier: LCSR (stopword detected)	peacekeeping mission	true
String Modifier: LC	in mali	false
String Modifier: LCSR (stopword detected)	mali	true

Table 3.1: Label Tokenization

Depicted is the sequence of strings that are looked up when using two modifiers and left-to-right tokenization on the term *United Nations Peacekeeping Mission in Mali*. Modifier *LC* stands for *lowercasing* and *LCSR* for *lowercasing and stopword removal*. Note that this is a simplified example with just two string modifiers; in the actual implementation, more string operations are applied before the last word is cut. The implemented linker will also never run a query twice for the same substring. Bold printed terms are found and returned.

Penalties for Incomplete Linking There are cases in which parts of a label cannot be found for example in *tubule macula* and in *macula lutea* both times only *macula* can be found using the ALOD Classic data set. If only the found concepts would be used to calculate the similarity between the concepts, a perfect score would be obtained because $sim(macula, macula) = 1.0$. However, this is not precise as this approach does not allow to discriminate between perfect matches

due to incomplete linking and *real* perfect matches. Therefore, a penalty factor $p \in [0, 1]$ is introduced that is to be multiplied with the final similarity score and which lowers the score for incomplete links; $p = 0$ indicates the maximal penalty, $p = 1$ indicates no penalty. The calculation of p is depicted in equation 3.1:

$$p = 0.5 * \frac{|Found\ Concepts\ L_1|}{|Possible\ Concepts\ L_1|} + 0.5 * \frac{|Found\ Concepts\ L_2|}{|Possible\ Concepts\ L_2|} \quad (3.1)$$

where L_1 is the label of the first concept and L_2 is the label of the second one; $|Found\ Concepts\ L_i|$ is the number of tokens for which a concept could be found (minus stopwords) and $|Possible\ Concepts\ L_i|$ is the number of tokens of the label without stopwords.

The penalty factor introduced is a relative measure: If only one part of a two-token label cannot be found, the penalty component for this label is $\frac{1}{2} = 0.5$ whereas if only one part of a four-token label cannot be found, the penalty component for that label is $\frac{1}{4} = 0.25$. Given the example $L_1 = tubule\ macula$ and $L_2 = macula\ lutea$ where only *macula* can be found both times, the similarity score with penalty factor (sim_p) is:

$$\begin{aligned} sim_p(L_1, L_2) &= (0.5 * \frac{1}{2} + 0.5 * \frac{1}{2}) * sim(macula, macula) \\ &= (0.25 + 0.25) * 1 \\ &= 0.5 \end{aligned}$$

Note that if there would be $L_3 = macula$ and $L_4 = macula$, the penalty score would equal 1 and, consequently, the similarity would be:

$$\begin{aligned} sim_p(L_3, L_4) &= (0.5 * \frac{1}{1} + 0.5 * \frac{1}{1}) * sim(macula, macula) \\ &= (0.5 + 0.5) * 1 \\ &= 1.0 \end{aligned}$$

As it can be seen from the examples above, by using the penalty factor, perfect matches due to incomplete linking can be discriminated against real perfect matches.

3.3 Basic Features

3.3.1 ALOD Features based on SPARQL

Based on the ALOD data set, features have been derived which can be used in the matching process. Feature generators have been implemented which can calculate

a numerical feature given two URIs. The features presented in the following are based on SPARQL queries and basic mathematical operations implemented in Java.

Number of Broader Concepts This feature generator counts the number of direct broader concepts of two URIs which shall be compared and calculates a similarity measure based on the absolute difference between the two values. The underlying assumption is that similar concepts have a similar amount of broader concepts in the data set. The similarity calculation is depicted in equation 3.2.

$$sim(c_1, c_2) = \frac{1}{1 - \left| |broader_{c_1}| - |broader_{c_2}| \right|} \quad (3.2)$$

where c_1 and c_2 refer to the concepts used and $|broader_{c_i}|$ refers to the number of broader concepts of concept i .

Number of Narrower Concepts This feature generator counts the number of direct narrower concepts of two URIs which shall be compared and calculates a similarity measure based on the absolute difference between the two values. The underlying assumption is that similar concepts have a similar amount of narrower concepts in the data set. The similarity calculation is depicted in equation 3.3.

$$sim(c_1, c_2) = \frac{1}{1 - \left| |narrower_{c_1}| - |narrower_{c_2}| \right|} \quad (3.3)$$

where c_1 and c_2 refer to the concepts used and $|narrower_{c_i}|$ refers to the number of narrower concepts of concept i .

A Has Broader Concept B This feature generator returns *true* if B is a broader concept, i.e. a hypernym, of A and *false* otherwise. Starting from concept A, broader concepts are retrieved and it is checked whether B is among them. If this is the case, the algorithm terminates; otherwise, broader concepts of broader concepts are retrieved and checked again until there are no broader concepts anymore. Pseudocode is given in algorithm 1. For reasons of performance, it makes sense to limit the number of broader concepts of broader concepts which shall be retrieved. Therefore, the implementation also allows to set a limit of how many steps upwards in the tree are allowed (this is referred to as *level* here). As it is expensive to load broader concepts and to loop over them, the actual implementation has been optimized to generate a SPARQL ASK query for each level which will return with a boolean. When switching the two input parameters, this feature can be converted into two features: *A Has Broader Concept B* and *B Has Broader Concept A*. For

equivalence relations, the direction of comparison does not matter as the function is symmetric, i.e., $sim(A, B) = sim(B, A)$. Therefore, the two features can be combined into one, named *OneHasOtherAsBroaderConcept* henceforth:

$$OneHasOtherAsBroaderConcept(A, B) = AHasBroaderConceptB(A, B) \vee BHasBroaderConceptA(A, B) \quad (3.4)$$

Algorithm 1 A Has Broader Concept B

```

broaderConcepts ← getBroaderConcepts(A)
for c ∈ broaderConcepts do
  if c == B then
    return true
  end if
end for
newBroaderConcepts ← broaderConcepts
while newBroaderConcepts ≠ ∅ do
  newNewBroaderConcepts ← ∅
  for c ∈ newBroaderConcepts do
    newNewBroaderConcepts.add(getBroaderConcepts(c))
  end for
  newBroaderConcepts ← newNewBroaderConcepts
  for c ∈ newBroaderConcepts do
    if c == B then
      return true
    end if
  end for
end while
return false

```

Broader Vector Space The feature generator takes the resources which shall be compared and follows the broader links within the graph. From the gained concepts a vector is built. Then, the Euclidean Distance between the two concept vectors is calculated and used as feature. The generator can be configured by setting the following parameters:

- **LEVEL**
Defines how many *hops* are allowed, i.e., up to which depth the graph is followed starting from the resource for which the vector shall be calculated.

- **LIMIT**
Defines how many broader concepts per level and node are retrieved. By default, all concepts are sorted by confidence and the top *LIMIT* concepts are selected and added to the vector (given that they fulfill all other requirements).
- **ELEMENT BASE VALUE**
Defines which value is used for an element in the vector space. Available options are *FIXED* (use 1.0) or *CONFIDENCE* (use not 1.0, but the confidence provided by the ALOD data set to build the vector).
- **DECAY FACTOR**
When *walking up the tree*, the concepts get broader, more abstract and more erroneous. Direct broader concepts are the most important ones. Therefore, a *DECAY FACTOR* is defined. The weight contribution to the vector shrinks exponentially as defined by the factor. The weight is calculated as:

$$w'_i = w_i * DECAY FACTOR^{CURRENT LEVEL-1} \quad (3.5)$$

where the *CURRENT LEVEL* is defined as the number of hops the concept is away from the starting resource.

In appendix C.4, an extensive example with a concrete calculation is given for illustrative purposes.

It is noteworthy to mention that this feature is very expensive to calculate due to the exponential number of broader concepts to retrieve with an increasing level.

Broader Overlap For this feature, all broader concepts of the concepts to be compared are obtained. The set similarity measures *Jaccard* (see equation 3.6 and more explanations in 3.3.2) and *Dice Coefficient* (see equation C.11 and appendix C.11 for more details) are implemented to calculate the similarity of overlapping concepts. It is also possible to limit the set of retrieved concepts by setting a minimal confidence threshold or by limiting the number of concepts that shall be obtained. For the latter option, the top k concepts sorted according to confidence are used. Furthermore, it is possible to include broader concepts of broader concepts in the calculation process by setting a level parameter. In figure 3.2, the top three broader concepts for J. K. Rowling²² and J. R. R. Tolkien²³ are given for a level of 1. Their similarity using the Jaccard coefficient would be $\frac{1}{3+3-1} = 0.2$ in this example.

²²see http://webisa.webdatacommons.org/concept/j.k_rowling_

²³see http://webisa.webdatacommons.org/concept/j.r.r_tolkien_

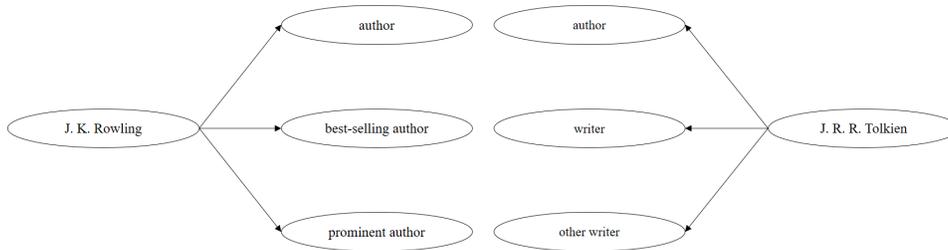


Figure 3.2: Overlapping Concepts Example

Depicted are the top three broader concepts of J. K. Rowling and J. R. R. Tolkien in ALOD Classic.

Narrower Overlap The *narrower overlap* feature is implemented analogously to *broader overlap*: Given a certain level and a minimum confidence, all narrower concepts of both URIs are retrieved. The set similarity measures, *Jaccard* and *Dice Coefficient*, are implemented to calculate the similarity of overlapping concepts.

3.3.2 Further Features

In addition to the features based on the ALOD data set introduced above, further features are implemented which are presented in the following.

Levenshtein on Label Based on the label of two matched concepts, traditional edit-distance measures can be applied. For this thesis, the Levenshtein algorithm [109] is used and available as feature. The algorithm counts the minimal number of character insertions, deletions, and substitutions required to transform one String into the other [30, p. 103]. This number can be length-normalized and transformed into a similarity measure in the range [0,1] according to equation C.9 given in the appendix (more information on Levenshtein is also given there).

Jaccard on Label The Jaccard coefficient is a common set similarity measure that can also be used as n-gram-based string comparison measure: The strings of the concepts to be compared are transformed into n-grams. Here, n is to be defined by the user; $|c_1|$ and $|c_2|$ are the number of n-grams of string 1 (s_1) and of string 2 (s_2) respectively. The similarity is based on the number of common n-grams (c_{common}) and is calculated as stated in equation 3.6. [30, pp. 106-107]

$$sim_{jaccard}(s_1, s_2) = \frac{c_{common}}{|c_1| + |c_2| - c_{common}} \quad (3.6)$$

3.4 ALOD2Vec Feature

The features presented so far allow for propositionalization based on SPARQL queries as well as basic mathematical operations. Besides those, a new approach which is based on the RDF2Vec method presented in subsection 2.5.1 has been implemented and evaluated: *ALOD2Vec*.

3.4.1 Random Walk Generation

Ristoski et al. present two different approaches to walk generation: (1) Random Walks and (2) Weisfeiler-Lehman Graph Kernels. The authors note that the latter approach "[does] not scale well to large-scale knowledge graphs, such as DBpedia or Wikidata" [147, p. 22]. This is due to the exponential growth of walks with increasing depth [39, p. 30]. Given the fact that the data set under consideration in this thesis is much larger than regular data sets, the random walk approach is pursued. In a follow-up paper [32], biased walks were presented for the first time. Here, the idea is to prefer meaningful walks by considering edge weights. In the scope of this thesis, a biased walk approach is also implemented.

The original RDF2Vec implementation is not optimal for the ALOD data set because of idiosyncratic properties:

1. The ALOD data set is organized in separate SPARQL graphs; the original implementation cannot handle those.
2. The ALOD data set can easily have a couple thousand object properties for one concept. In the original implementation, random walks are generated by calculating the Cartesian product for multiple hops. While that might work for other knowledge graphs, this approach leads to a memory explosion for the ALOD data set and is neither time efficient nor easily feasible.²⁴
3. Compared to other knowledge graphs, the WebIsA data set only has one semantically meaningful object property: `skos:broader`. The inclusion of the property to the walks is not adding additional value.

In order to overcome the obstacles mentioned above, the approach was reimplemented and adapted. Two different options are available:

Jena-Based Walk Generation The first step of walk generation is to load the data.²⁵ As the n-quads files are not clean (some IRIs contain spaces and cannot

²⁴Experiments were performed on a machine with 125 gigabytes of RAM.

²⁵The data is available here: <http://webisa.webdatacommons.org/>

be loaded into Jena out-of-the-box), a preprocessing of the files to be loaded is required. This is implemented as a Java program using regular expressions. The preprocessed file can afterwards be loaded into Jena TDB and be queried. The advantage of this approach is a relatively low memory requirement. The disadvantage is its low processing performance and large front-up investment in order to preprocess the graph and load it into TDB.

Memory-Based Walk Generation The Jena-based approach is very slow due to the disk access of TDB and SPARQL-induced overhead. Therefore, a memory-based approach has also been implemented. The n-quads file is first stripped from any non-relevant information (such as meta data concerning the patterns or pay-level domains) and rewritten to disk. Afterwards, the relations are loaded into performance-optimized data structures and are held in memory. Here, no SPARQL is used. This approach is more than a magnitude faster compared to the approach presented above but has hefty memory requirements: In order to hold the ALOD XL relations in memory, at least 80 GB of RAM are required.

Given a graph $G = (V, E)$, the original walk pattern for each vertex $v \in V$ in the second iteration is $v_r \rightarrow e_{1_i} \rightarrow v_{1_i}$ where v_r is the root node and $e_{1_i} \in E(v_r)$. The algorithm then continues until the number of specified iterations d is reached. [147, p. 4] As the edges are always equal to `skos:broader`, this information is discarded and the pattern for two iterations in ALOD2Vec is as follows: $v_r \rightarrow v_1 \rightarrow v_2$ where $(v_r \text{ skos:broader } v_1) \wedge (v_1 \text{ skos:broader } v_2)$ holds. The pattern can be continued for as many iterations as desired. An example for a random graph of depth 2 on the ALOD Classic data set would be *president* \rightarrow *political leader* \rightarrow *public figure*. The walks themselves resemble those generated in node2Vec even though the walk generation is different.

For the Jena-based approach, the query process itself has also been rewritten compared to the original implementation: In a first step, all concepts are collected for which walks shall be created. Then – rather than creating Cartesian products – for each segment of the walk a concept is drawn. The drawing can be configured to be biased in the sense that hypernyms which received a higher confidence are more likely to be drawn which can be seen in the Java query generation method of listing 3.2: The confidence is combined with a random element so that high-confidence concepts are drawn more often but not always. Compared to the original walk creation method, this approach scales with linear time complexity with respect to the number of desired walks per concept and the desired depth. The memory-based generator works in a similar fashion, as depicted in listing 3.3, and is also capable of generating biased walks. In both cases, the walk generation is multi-threaded so

that the runtime improves on machines with many processing units. The number of threads can be set dynamically according to the capabilities of the hardware.

```

1    /**
2    * Generate the query to obtain a random
      broader concept.
3    * Note: The probability is greater for
      high-confidence hypernyms.
4    * @param concept: The concept IRI for which a
      broader concept shall be retrieved.
5    * @return Query in String representation.
6    */
7    public String
      generateQueryForRandomBroaderConcept(String
      concept){
8        return PREFIX_SKOS + PREFIX_ISAO +
9            "SELECT ?e WHERE\n" +
10           "{ GRAPH ?g {<" + concept + ">
              skos:broader ?e .}\n" +
11           "?g isao:hasConfidence ?c .\n" +
12           "BIND(RAND()*?c AS ?rank)} ORDER
              BY ?rank LIMIT 1";
13    }

```

Listing 3.2: Random Broader Concept Generator Method: SPARQL-Based

```

1    /**
2    * Returns a random broader concept of the
      given concept with a higher probability of
      high-confidence concepts
3    * to be drawn.
4    * @param concept: The concept for which the
      hypernym shall be retrieved.
5    * @return The random broader concept in
      String representation.
6    */
7    public String drawRandomConcept(String
      concept){
8        String result = "";
9        float greatestConfidence = 0f;
10       float currentConfidence = 0f;

```

```

11     Random rand = new Random();
12     ArrayList<StringFloat> broader =
13         broaderConcepts.get(concept);
14     if(broadener == null){
15         return null;
16     }
17     for(StringFloat s : broader){
18         currentConfidence = rand.nextFloat() *
19             s.floatValue;
20         if(currentConfidence >=
21             greatestConfidence){
22             result = s.stringValue;
23             greatestConfidence =
24                 currentConfidence;
25     }
26     }
27     return result;
28 }

```

Listing 3.3: Random Broader Concept Generator Method Memory-Based

Reverse Walks The random/biased walk generation algorithm presented so far has one disadvantage: While all nodes with outgoing edges appear in the set of walks with certainty, nodes with only ingoing edges are only present when they appear in some other random walk.

Furthermore, nodes with only ingoing edges likely appear in much fewer walks and the derived embeddings are likely less meaningful. In the data set at hand, however, there are many such nodes [153, p. 5]. Therefore, walks for hypernyms that do not appear as hyponyms were generated in reverse order: The concept *european organization*²⁶, for instance, has only narrower concepts. A resulting reverse walk of depth 2 might look like this: *european organization* ← *european union*. After the reverse walk generation, the walks are transformed so that they follow the format of the other sentences. Given the example above, the resulting walk looks like this: *european union* → *european organization*.

This notion is neither present in node2Vec nor in RDF2Vec.

²⁶see [http://webisa.webdatacommons.org/concept/european_](http://webisa.webdatacommons.org/concept/european_organization_)
[organization_](http://webisa.webdatacommons.org/concept/european_organization_)

3.4.2 Training of Embeddings

The walks are persisted in compressed files. Those are read by a Python program for the training step. An example of the ten closest concepts obtained using the SG-500 ALOD embeddings can be found in table 3.4.2.

The walk generation was successfully run for ALOD Classic and ALOD XL. However, embeddings could not be trained for the full ALOD XL data set because of very high memory requirements: The gensim library keeps the embeddings in memory. In order to do so, the memory requirement can be estimated as:

$$Memory = |concepts| * dimension * 8 \text{ bytes} \quad (3.7)$$

where $|concepts|$ refers to the number of concepts and $dimension$ refers to the desired dimension for the vectors.

The 8 bytes resemble the memory requirements for a double.²⁷ For 200,000,000 concepts and a dimension of 200, this means that at least $200,000,000 * 200 * 8 \text{ bytes} = 320 \text{ GB}$ would be required. In accordance with the available hardware, the process was started for the ALOD XL data set with a dimension size of 50. After several days, the embeddings were trained but a memory error occurred while writing those to a file. The approach was not pursued afterwards. Here, it can be seen that node2Vec and RDF2Vec have very high requirements when it comes to graphs with many nodes. The bottleneck is not the generation of paths but rather the training of the embeddings.

Embeddings Configuration In order to train the embeddings, the gensim library is used. For the training, the parameters of the original RDF2Vec publication have been chosen [147, p. 7]: *window size = 5*; *number of iterations = 5*; *negative sampling for optimization*; *negative samples = 25*; *average input vector* for CBOW.

The ALOD Classic embeddings were calculated with 100 walks per entity, depth 8 and enabled reverse-walk mechanism.

In order to evaluate whether a larger data set improves the results, a subset of the ALDO XL data set has been created in a different fashion compared to the original WebIsALOD paper: Given a specific concept c from the set of all concepts C , the subset criterion is:

$$dataset(n) = \{c \in C \mid (|c_{in}| + |c_{out}|) > n\} \quad (3.8)$$

²⁷The memory requirement is actually even greater because there is additional overhead: This calculation does not cover the vocabulary or the references of the vectors to the corresponding terms.

where $|c_{in}|$ represents the number of in-going edges and $|c_{out}|$ represents the number of out-going edges. The idea behind this selection criterion is that meaningful embeddings can only be trained, when the concept is involved in more than n relations.

For further experiments $n = 10$ has been set, meaning that each concept in that data set is involved in at least 10 relations, i.e., each node has at least a degree of 10. The resulting reduced XL data set, referred to in the following as *ALOD XLR*, holds a little more than 5 million concepts and is more than three times as large as *ALOD Classic*. However, the data set is still small enough to train embeddings. In order to save disk-space, only 200-dimensional embeddings have been trained for this particular subset.

swap		eu		pancreas		a380	
ALOD	GOOG	ALOD	GOOG	ALOD	GOOG	ALOD	GOOG
total return swap	swaps	international forum	european	acute pancreatitis	liver	singapore airlines	
clearing requirement	swapping	quartet member	uk	chronic pancreatitis	small intestine	harrier jump jet	
major swap participant	swap	political project	europa	elastase	intestine	china eastern	
option on future	swapped	supra-national body	ce	secretory organ	kidneys	efficient aircraft	
basis swap	exchange	donor of development assistance	te	hereditary pancreatitis	blood vessels	a340	
interest rate swap	transfer	use interchangeably	ted	pancreatitis inflammation	bone marrow	u-2 spy plane	
clear swap	exchanging	supranational state	ca	pancreatitis	tumor	boeing b-17 flying fortress	
retail offer	sell	foreign policy chief	ar	disorder of the pancreas	pancreatic	end of next year	
interest rate swap agreement	swapping	foreign counterpart	ut	chymotrypsin	cancerous cells	ne boeing 787 dreamliner	
foreign exchange swap	deal	mara jade				airbus 380	

Table 3.2: ALOD2Vec: Top 10 Closest Concepts

The ALOD column displays the 10 closest concepts using the CBOW-200 ALOD Classic embeddings. As a comparison, the 10 closest concepts for the model by Google are given (column GOOG). Note that a380 is not a known concept to the GOOG model; neither are airbus a380 or a 380. Here, one advantage of ALOD2Vec becomes apparent: Its large number of concepts. The Google model is publicly available: <https://code.google.com/archive/p/word2vec/>

3.4.3 Hardware

For the walk generation and the learning of embeddings a server with 40 cores à 2.60Ghz and 128GB of RAM running SUSE Enterprise Linux was used.²⁸

3.5 Feature Selection and Weighting

3.5.1 Selection and Weighting Process

When combining different features, a strategy is required in order to still being able to make decisions. One way is a *similarity aggregation strategy*. An aggregated similarity value ($sim_{aggregate}$) can be calculated as follows [1, p. 4]:

$$\begin{aligned} sim_{aggregate}(c_1, c_2) &= \sum_{i=1}^h w_i * sim_i(c_1, c_2) \\ &\text{subject to } \sum_{i=1}^h w_i = 1 \end{aligned} \quad (3.9)$$

where c_1 and c_2 are the concepts for which the similarity shall be calculated, h is the number of features, sim_i is the value of the i^{th} feature, and w_i is the weight for the i^{th} feature.

By applying equation 3.9, a single confidence value can be assigned to a correspondence even though multiple features are used. In order to learn good weights, machine learning techniques, like simple regressions, can be used. Some techniques can set weights w_i to zero and, thereby, select only relevant features.

3.5.2 Monosemous Synonymy Gold Standard (MSGs-1234)

A prerequisite for every supervised learner, however, is a gold standard of labeled instances. In the context of this thesis, there are requirements for a good gold standard: Matching using the ALOD data set is label-based; therefore, a good property between two labels for learning is required. In this case, synonymy (see 2.1.2) was chosen as a desirable property. When exploiting pure labels, certain problems arise, namely words carrying multiple meanings (polysemy, see 2.1.2) which can lead to situations where two labels are equal but the encoded sense is different (homonymy, see 2.1.2). The gold standard needs to be free of such grey-scale cases and at the same time allow for good corner cases which are in this case, for instance, related but not synonymous words.

Due to the lack of existence, the author of this thesis created a gold standard which contains 1234 noun-noun-pairs together with a binary indicator stating

²⁸The server has been provided by SAP SE.

whether the nouns can be used synonymously in a strong-from interpretation, i.e., whether they are usable interchangeably in any context (see subsection 2.1.2). The gold standard does contain polysemous words for negative annotations but all positive examples are monosemous. As a starting point, all 541 nouns of the McRae et al. feature norms [115], which are publicly available²⁹, were used. The data set was chosen because all words are visually perceivable, the degree of abstraction is low, and the words are likely to be monosemous. All nouns in the gold standard can be found on WordNet [134]. A positive match between two words was annotated when the terms in question are members of the same synset(s) and are exposed to the synset(s) in question exclusively according to WordNet. The words *doorknob* and *doorhandle*, for instance, are both used in one synset (which is their only one) with the explanatory text stating "a knob used to release the catch when opening a door" [133]. Therefore, the two concepts are added to the gold standard as a positive example.

Negative matches were chosen by selecting related nouns according to the word2vec methodology [117] utilizing Google's publicly available³⁰ pre-trained entity vectors. The criterion for a negative annotation was that the related word does not occur in any synset of the other word. The Python code used to obtain the most related words is given in the appendix (listing A.2). All annotations were checked by a second reviewer and only added when consent existed concerning the word pair and its label. In total, there are 360 positive and 874 negative annotations. The chosen structure of the gold standard makes it possible that MSGS-1234 can also be viewed as a regular synonymy gold standard with very strong synonymy ties and can also be used in other contexts. The gold standard was made publicly available on GitHub³¹ and can also be found on the CD enclosed to this thesis in digital form.³² Results in the context of ontology matching are reported in section 4.5.

3.6 Matcher Details

3.6.1 Implemented Matching Restrictions

As outlined in 2.3.1, multiple matching restrictions are possible. In the scope of this thesis, multiple strategies have been implemented and evaluated which are presented in the following.

²⁹see <https://sites.google.com/site/kenmcraelab/norms-data>

³⁰see <https://code.google.com/archive/p/word2vec/>

³¹see <https://github.com/janothan/MSGS-1234>

³²Due to the sheer size the gold standard is not available in the appendix.

Plain Strategy (N:M) This is the most basic strategy. Here, all correspondences above a certain threshold are added to the resulting alignment.

Best Match (1:N) Given two ontologies O_1 and O_2 to be matched, the plain strategy described above might match multiple resources of O_1 to multiple resources of O_2 . The *Best Match* strategy takes the set of correspondences S_{1_i} for each resource $R_{1_i} \in O_1$ and only adds the correspondence with the highest confidence to the resulting alignment. If there are multiple highest ranked correspondences in S_i , a referee function is used. In this case, the correspondence with the best Levenshtein edit similarity is chosen. If there still is a draw, i.e. the edit similarity is also identical, a random draw is performed. The pseudocode can be found in algorithm 2. Note that the algorithm uses the generic term *resource* which leaves open whether classes, data properties, and object properties are referred to as separate sets or as combined sets.

Algorithm 2 Best Match Strategy (1:N)

```

alignment ← newSet()
resource1 ← ∅
resource2 ← ∅
for resource1 ∈ Ontology1.resources do
  maxConfidence = 0
  bestResource2 ← ∅
  for resource2 ∈ Ontology2.resources do
    similarityResult = similarity(resource1, resource2)
    if similarityResult > maxSim then
      maxConfidence = similarityResult
      bestResource2 = resource2
    else if similarityResult == maxSim then
      bestResource2 = referee(resource2, bestResource2)
    end if
  end for
  alignment.add(resource1, bestResource2, maxConfidence)
end for
return alignment

```

Best One-to-One (1:1) Note that when applying *best match* it is still possible that multiple r_{1_i} refer to the same $r_{2_i} \in O_2$. The approach presented here selects the best match by asserting at the same time that each element in O_1 refers to an

element in O_2 that is not used in another correspondence. While it is not trivial to find the optimal solution, a simple greedy heuristic can be applied [30, pp. 158-159] which is formulated as pseudocode in algorithm 3: Here, all similarity values are sorted in descending order and added to the final alignment in a top-down fashion, given that the two resources to be mapped were not mapped before.

Algorithm 3 Best One-to-One Heuristic (1:1)

```

alignment ← newSet()
mappingsBefore ← newList()
alreadyMapped ← newSet()
resource1 ← ∅
resource2 ← ∅
for resource1 ∈ Ontology1.resources do
  for resource2 ∈ Ontology2.resources do
    confidence = similarity(resource1, resource2)
    mappingsBefore.add(resource1, resource2, confidence)
  end for
end for
sortDescending(mappingsBefore)
for mapping ∈ mappingsBefore do
  if (mapping.resource1 ∉ alreadyMapped) ∧
  (mapping.resource2 ∉ alreadyMapped) then
    alignment.add(mapping)
    alreadyMapped.add(mapping.resource1)
    alreadyMapped.add(mapping.resource2)
  end if
end for
return alignment

```

3.6.2 Handling of Sub-Concepts

As stated earlier, the similarity between two concepts can be calculated straightforwardly. However, if a concept consists of many sub-concepts, for example when the concept *Council of the European Union* is mapped to the concepts *Council* as well as *European Union* and shall be compared with the concept *European Union*, a processing rule has to be implemented. Two options are implemented: (1) *Average* and (2) *Best Average*. Both are described in the following.

Average The simplest option is to calculate the cross product of the sub-concepts and average their similarity:

$$sim_{average} = \frac{\sum_{i \in c_1}^{c_1} \sum_{j \in c_2}^{c_2} sim(c_{1_i}, c_{2_j})}{|c_1| * |c_2|} \quad (3.10)$$

where c_1 and c_2 represent two individual concepts and c_{1_i} respectively c_{2_j} represent the i^{th} and j^{th} sub-concept of c_1 and c_2 ; $|c_1|$ and $|c_2|$ are the number of subconcepts of c_1 and c_2 .

Best Average The approach above can lead to the situation that perfect matches do not receive a score of 1.0 because, using the example stated above, matching *Council of the European Union* with itself would lead to *Council* not only being matched with itself but also with *European Union* which is most likely not a perfect match. Therefore, the *Average* approach is extended:

$$sim_{average} = \frac{\sum_{i \in c_1}^{c_1} Max_{j \in c_2}^{c_2} sim(c_{1_i}, c_{2_j})}{|c_1|} \quad (3.11)$$

where c_1 is the concept with more tokens.

This is a hard requirement in order to avoid full scores given to longer concepts that contain the other, shorter concept, i.e., $c_2 \subset c_1$. An example for such a situation would be *muscle* and *smooth muscle tissue cell*. If *muscle* were the first concept, it would only be matched with the *muscle* part of the other concept and, hence, receive a score of 1.0 according to equation 3.11.

In this thesis, multiple sub-concepts occur frequently and *Best Average* is used to handle them. This ensures that perfect matches receive a score of 1.0.

3.6.3 Handling of Multiple Textual Parts

Some ontologies assign multiple labels to classes (e.g. the NCI Thesaurus Ontology), others do not have labels at all (e.g. the EDAS Conference Ontology). Furthermore, in some cases there might be additional comments or definitions. In order to handle all such cases, a mechanism is required to calculate the similarity among multiple labels. For the matcher in this thesis, a score matrix is used (similar to the one in WeSeE-Match [128, p. 215]): In a first step, all values of `owl:AnnotationProperty` and its sub classes are retrieved. This includes `owl:versionInfo`, `rdfs:label`, `rdfs:comment`, `rdfs:seeAlso`, `rdfs:isDefinedBy`. Additionally, the IRI fragment is added. Each retrieved String from one ontology is compared to each retrieved String

from the other ontology and the maximum similarity is returned. This is also depicted in equation 3.12:

$$\text{sim}(e_1, e_2) = \max_{i,j \in (\text{annotations} \cup \text{fragment})} \text{sim}(str_i(e_1), str_j(e_2)) \quad (3.12)$$

where e_1 is an element from one ontology and e_2 is an element from another ontology, sim is a similarity function which calculates the similarity accepting two string representations and $\text{str}(e)$ is one string element from the union of annotation properties and fragments of element e . A concrete calculation example is given in table 3.3.

		e_1	
		MA0000270	eyelid tarsus
e_2	NCI_C33736	0.022	0.0
	Tarsal_Plate	0.001	0.037

Table 3.3: Score Matrix Example

This is a real-world example from the OAEI Anatomy data set using the narrower overlap on the ALOD XL endpoint. The bold printed labels are the annotations found for e_1 and e_2 , respectively. The assigned score for $\text{sim}(e_1, e_2)$ is $\max(0.022, 0.0, 0.001, 0.037) = 0.037$.

3.6.4 Performance-Based Enhancements

Compared to non-Web-based approaches, one disadvantage of the ALOD-based matcher is runtime performance. Similar to WikiMatch [74, p. 44], the bottleneck for SPARQL-based features is the time it takes to send out requests and to retrieve data. However, all approaches presented in this thesis scale linearly in terms of search requests as opposed to other approaches such as the *Google Distance* [31]³³ where the search requests scale quadratically. Hence, the similarity computation is scalable even though it is slow. Furthermore, as the comparison of two concepts is an independent task, there is much potential for parallelization.

The ALOD2Vec approach does not have network induced runtime problems but rather suffers from the size of the vector corpus.

In order to improve runtime performance the following methods are applied:

³³The basic idea behind the Google Distance is that two semantically similar concepts appear on the Web more often together and consequently lead to more results retrieved by the Google search engine. However, this approach requires (among others) a search request for each two concepts that shall be compared. [31, p. 5]

- A *string filter* is implemented which filters out trivial matches (see architecture overview in figure 3.1). This allows to filter out direct string matches without querying external resources for those concepts for linking and similarity calculation. In its basic form, this does not change the results in any way as the approach presented in this thesis relies solely on textual representations. However, if the ALOD approach shall be combined with other string-based approaches, the filter can be extended, e.g. by adding a Porter-stemmer [131] or a lemmatizer.
- The string filter is multi-threaded. This is possible because comparisons are independent of each other. At the beginning of the process, the elements of one ontology are divided into subgroups that are distributed to threads which compare the subsets against all elements of the other ontology. This significantly improves the runtime on very large ontologies without sacrificing the quality of the result. The number of threads can be set dynamically according to the capabilities of the hardware. In order to not cause overhead, multi-threading is automatically disabled when matching smaller ontologies.
- The linking process itself is expensive for very long texts as multiple concepts are derived from the textual representation (see 3.2). Therefore, text can be cut after x characters. For the experiments in the following chapter, $x = 100$ has been chosen.
- Fragments do not always encode valuable information. In the OAEI Anatomy data set, for instance, fragments receive an ID such as NCI_C12220 or MA_0000003. These kind of strings slow down the comparison process without adding any value. Therefore, fragments which contain more number characters than half of the total characters of the String are ignored.
- As the bottleneck for SPARQL-based features are queries, a buffer has been implemented so that no query will be sent out twice to the Web service. Rather than associating a full query with its result, a `SparqlService` class exposes certain query services and uses different buffers for the required data structures. At no point `ResultSet` objects are buffered, but instead lightweight `HashMaps` with the minimal information required for the application are cached.
- The same buffering concept is also implemented for the label to concept linkers in order to avoid performing the linking process twice. This is also used for non-SPARQL approaches.

- All buffers can optionally be persisted so that different configurations can be tried out in a row.

The ALOD2Vec approach does not rely on Web-queries and performs much better in terms of runtime. As vector comparisons and performing the linking process are still more expensive than string comparisons and loads from a HashMap, the string filter as well as the buffers were kept for this approach where applicable. Another challenge arises from the embeddings: The size of the files holding the embeddings is considerable – 2GB for the 200-dimensional embeddings and 5GB for the 500-dimensional embeddings on ALOD Classic.³⁴ Problems on smaller PCs arise when there is only limited memory available. Furthermore, loading a 5GB large HashMap requires a considerable start-up time. Therefore, besides the memory approach, another option has been implemented: Using a helper program, the embeddings can be loaded into a SQLite database. The usage of the embedded RDBMS reduces the startup-time, requires far less memory, and yields good performance. As disk-access is slower than memory-access, queries are buffered to increase the overall performance. Due to a library incompatibility with the SEALS framework, the same approach is also implemented using the MapDB library.³⁵

3.7 Implementation Details

This section quickly covers the accompanying materials of this thesis which are stored on the enclosed CD and discusses actions taken to ensure code understandability and quality.

3.7.1 Projects

The accompanying CD contains all coding artifacts developed in this thesis. It is divided into five projects which are separated in different folders:

- `main_project`
This directory contains the main project. It is implemented in Java 8 using maven³⁶ as dependency manager. Different applications (see next section) are grouped in packages.
- `oaei_2018_project`
This directory contains a fork of the main project. More specifically, it contains the the final matcher along with the required dependencies.

³⁴When loading such a file into memory, the required memory here is even greater.

³⁵It turned out that the MapDB library is faster than SQLite. However, its capabilities are not as rich and the library might not be sufficient for future enhancements.

³⁶see <https://maven.apache.org/>

- `oaei_2018_seals`
This directory contains the SEALS package that will be submitted for the OAEI 2018 campaign.
- `alod2vec_learning`
This directory contains the Python code used to derive the embeddings. Note that the walk generation is implemented in the main project.
- `powerdesigner_extension`
This directory contains the implemented extension which allows to derive an ontology from a given PowerDesigner data model and also to export mappings in the Alignment API XML format. The code is written in VBS and packaged as PowerDesigner Extension (`.xem` file).

In addition, the following folders contain non-code artifacts:

- `msgs`
This directory contains the MSGS-1234 synonymy gold standard that was described in subsection 3.5.2.
- `thesis_written`
This directory contains the digital PDF version of this thesis.
- `thesis_presentation`
This directory contains the slides used to present the contents of this thesis.

3.7.2 Code Quality and Documentation

The majority of the implementation is written in Java. In order to ensure a high quality of code, more than 100 unit tests were implemented using the JUnit Test Framework³⁷.

In addition, to allow for readability as well as reusability, the coding is documented using JavaDoc. For code parts written in Python and VBS, regular comments are used to ensure clarity. Furthermore, packages and directories contain markdown files (named `README.md`) that explain the content and purpose of the current directory.³⁸ If the package contains an application, the documentary markdown file also explains how to run it.

For any details about the individual artifacts, refer to the corresponding `README.md` files as they cannot be fully explained here.

³⁷see <https://junit.org/junit5/>

³⁸This format was also chosen because GitHub automatically displays the `README.md` when navigating into a directory containing such a file. Thereby, understandability is improved when viewing the project on this platform.

3.7.3 Applications

The main project contains an `application` directory which again contains all classes that can be run. Subfolders structure the applications by their overall topic. Runnable programs can be recognized by the suffix `Application`.

All programs are written to be run from within an IDE which means that parameters have to be set within the code rather than when calling in the command line.³⁹ This is done in order to better explain the parameters within the code. Applications are implemented in a way so that there are only (documented) parameters to be set but there barely is application logic (except for orchestration calls) to ensure a maximum of clarity. This way, applications only act as a wrapper to the underlying logic which is stored in folders named `controller`.

³⁹Note that with very few clicks every application can be transformed to a command line program.

Chapter 4

Experiments

4.1 Overview of the Experiments

Experiments have been conducted to further understand the data and to evaluate the overall approach. The subsequent section (4.2) focuses on the ALOD data set and presents helpful statistics. In section 4.3, the concept coverage of the ALOD data set is benchmarked against DBpedia on nine different data models. Section 4.4 covers experiments performed to test the capability of the ALOD data set to capture semantic knowledge. Therefore, three publicly available gold standards are used and the ALOD features are benchmarked against other approaches. In section 4.5, the results of regressions on MSGS-1234 are presented. Eventually, the performance for ontology matching is benchmarked using publicly available data sets as well as an SAP-specific data set. The last section (4.7) covers a concrete use case in a business environment.

4.2 Descriptive Analysis of the ALOD Data Set

4.2.1 Data Set Size

As discussed earlier, the ALOD Classic data set is significantly smaller than the original data set due to filtering according to the number of pay-level domains and patterns. As a result, the classic data set's size equals only a tenth of the original data sets size when it comes to file size. Measured by the number of contained relations, the original data set is 36 times larger; measured by the number of concepts, it is 141 times larger. When comparing the average relations per concept (7.76 in ALOD Classic, 1.89 in ALOD XL), one can see that the ALOD Classic data set filters out concepts with few relations. Table 4.2.1 contains a detailed comparison

of the two data sets.

	ALOD Classic	ALOD XL
GB zipped	10 GB	75 GB
GB unzipped	84 GB	851 GB
# of Concepts	1,510,980	212,155,729
# of Hypernymy Relations	11,721,537	400,533,808
Lines in N-Quads File	603,246,828	5,637,335,455

Table 4.1: Comparison of ALOD Classic and ALOD XL

The number of concepts, relations, and lines was obtained by running small Java programs on the zipped data sets available online.

4.2.2 Distribution of Relations

In order to better understand the knowledge graph, the distribution of relations was analyzed. Therefore, the number of relations $|r|$ per concept were calculated. Considering that the data set can also be viewed as a graph, $|r|$ represents the degree $d = |e_{in}| + |e_{out}|$ where $|e_{in}|$ are the number of ingoing and $|e_{out}|$ are the number of outgoing edges. In this case, the number of relations equals the sum of the number of hypernymy and hyponymy relations in which a particular concept is involved. Hence, each relation is counted twice: Once for the hyponym and once for the hypernym. By calculating the frequency, i.e. the sum of instances per $|r|$, one can obtain the distribution of relations.

When having a look at the distribution, one can see that it follows a power law. This is a common property of large networks that also accounts to the World Wide Web [106, pp. 120-121]; Barabási and Albert explain this phenomenon with the *preferential attachment process*: Vertices tend to link to well-connected vertices [8, p. 509].

Figures 4.1 and 4.3 depict the power-law like distributions for the two data sets: The frequencies are decreasing with the number of relations, the absolute maximum for both distributions is at $|r| = 1$. In order to better judge the distribution, figures 4.2 and 4.4 depict the distribution in the range $|r| \in [1, 30]$.

In ALOD Classic, concepts with 1 to 5 relations make up 85% of the whole data set.¹ The 1 to 10 range accounts for 91% of the set. On the other extreme, there are 578 concepts with more than 5,000 relations; the ultimate leader with 91,102

¹When browsing the data set in the web-based view, this cannot be easily concluded, as most *common-world* entities one usually looks for have many hypernymy relations. The analysis presented in this section shows that the first subjective impression can be misleading.

relations is *thing*².

The situation is similar for ALOD XL. Here, however, the distribution is even more extreme: Concepts with 1 to 5 relations make up 95% of the data set. The 1 to 10 range accounts for 97% of the whole data set. The concept involved in most relations (1,113,297) is again *thing*³. When comparing the distributions, one can see that the ALOD Classic data set favors concepts with more relations.

All numbers were obtained by self-written programs that operate on the gzipped n-quads files rather than the SPARQL endpoints due to performance and memory requirements.

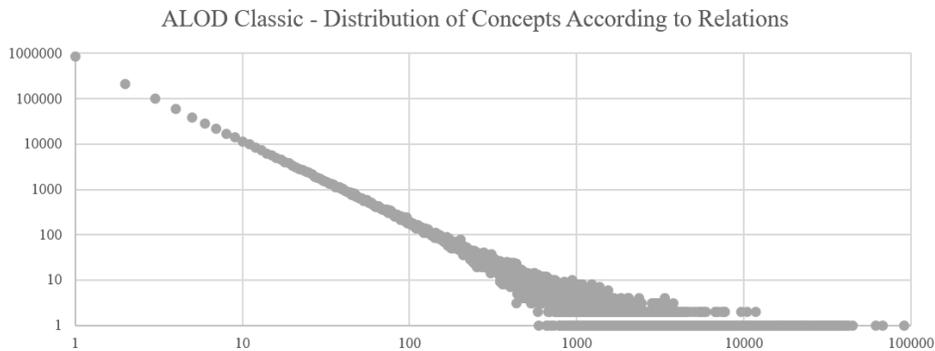


Figure 4.1: Distribution of Relations on ALOD Classic

The x-axis displays the number of relations in which a concept is involved, the y-axis displays the frequency per class. The scatter-plot is scaled on a log-log scale with base 10 in order to illustrate the power-law distribution.

²see http://webisa.webdatacommons.org/concept/_thing_

³XL IRI: <http://webisa.webdatacommons.org/concept/thing>

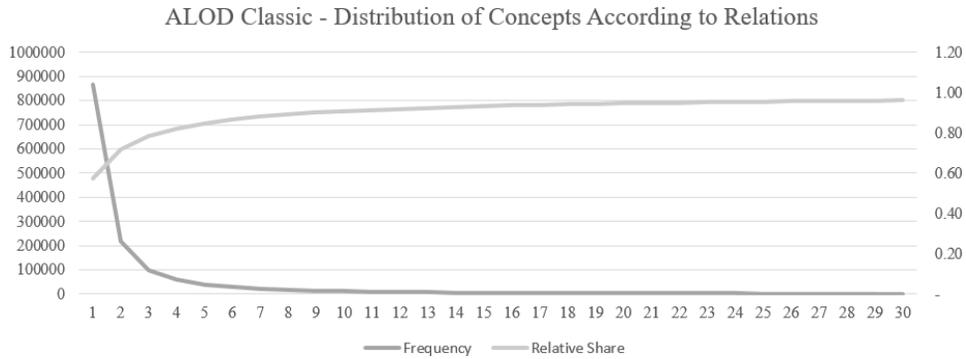


Figure 4.2: Distribution of Relations of ALOD Classic in the Interval $[1 - 30]$
 The upper, lighter gray line represents the cumulative relative share (scale is on the right y-axis). The frequency for $|r| = 30$ is 1581. The concepts in the given interval represent a 96% share of the data set.

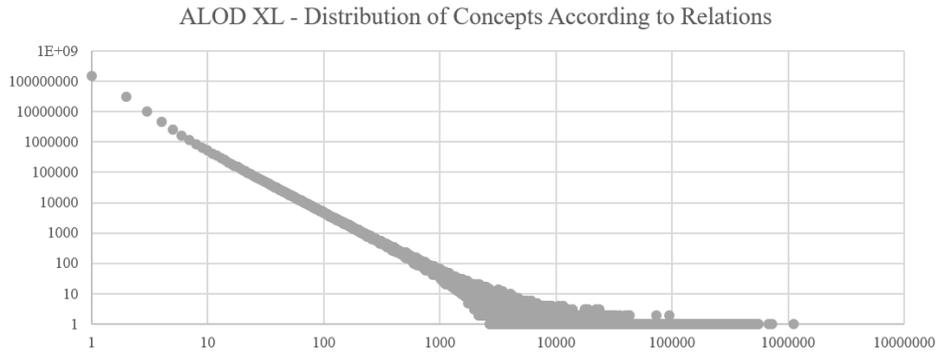


Figure 4.3: Distribution of Relations of ALOD XL
 The x-axis displays the number of relations in which a concept is involved, the y-axis displays the frequency per class. The scatter-plot is scaled on a log-log scale with base 10 in order to illustrate the power-law distribution.

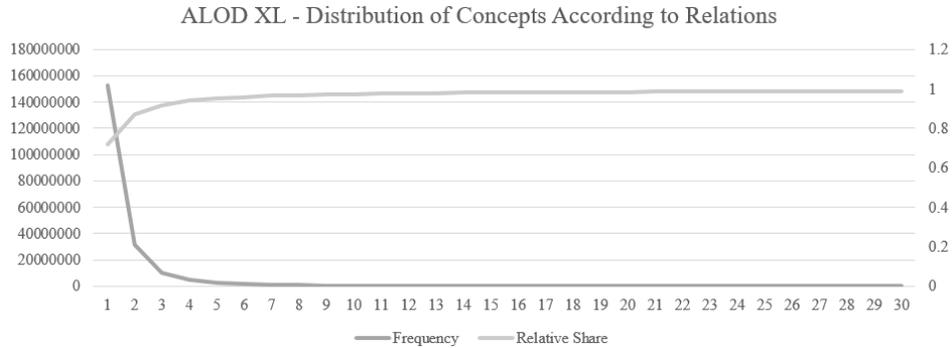


Figure 4.4: Distribution and Relative Share of Relations of ALOD XL in the Interval [1-30]

The upper, lighter gray line represents the cumulative relative share (scale is on the right y-axis). The frequency for $|r| = 30$ is 52,407. The concepts in the given interval represent a 99% share of the data set.

4.3 Coverage Calculations

To evaluate whether the WebIsA data set is suited for the task of ontology matching, nine data models have been chosen and the percentage of terms that can be matched to a WebIsA concept has been evaluated. The linker presented in 3.2 has been used to link data model entities and attributes (in case of relational data models) as well as classes and properties (in case of ontologies) to Web resources. For all data sets, English labels have been extracted. In total, 8 OAEI 2017 data sets have been used: *Anatomy*⁴, *Conference*⁵, *Biomed*⁶, *Disease and Phenotype*⁷, *University Admission*⁸, *Birth Registration*⁹, *Synthetic*¹⁰, and *Doremus*¹¹. In addition, SAP's conceptual financial service domain data model has been evaluated as well, using all names of entities, attributes, relations, and inheritances of the entity relationship diagram. In order to have comparative figures, the same process was used to link to three different endpoints: (1) DBpedia, (2) ALOD Classic, and (3)

⁴see <http://oaei.ontologymatching.org/2017/anatomy/index.html>

⁵see <http://oaei.ontologymatching.org/2017/conference/index.html>

⁶see <http://www.cs.ox.ac.uk/isg/projects/SEALS/oaei/>

⁷see <http://sws.ifi.uio.no/oaei/phenotype/>

⁸see <http://web.informatik.uni-mannheim.de/oaei/pm17/>

⁹see <http://web.informatik.uni-mannheim.de/oaei/pm17/>

¹⁰see http://islab.di.unimi.it/content/im_oaei/2017/

¹¹see http://islab.di.unimi.it/content/im_oaei/2017/

ALOD XL.

The evaluation is performed using two measures: (1) *whole-term coverage* and (2) *token coverage*. For the whole-term coverage, the texts are tokenized. A match is only counted if at least 80% of the tokens (minus stopwords¹²) can be mapped to the corresponding data set.¹³ The implemented calculator can handle concepts which consist of multiple tokens correctly; for example, the label *European Union* (two tokens) would be mapped to only one concept in ALOD, i.e. *European Union*¹⁴, but the algorithm recognizes that the concept consists of two tokens and counts this as a full match.

Results When comparing the ALOD Classic coverage with the DBpedia coverage, there is no clear superior approach (see table B.1 in the appendix): For some data sets, like *Doremus*, *FSDM*, or *Conference*, ALOD Classic performs better whereas for others, like *BioMed* or *Disease and Phenotype*, DBpedia has a better coverage. One general observation, though, is that while DBpedia and ALOD Classic do not perform well considering whole terms, the results are generally good when tokenizing the labels.

For the ALOD XL data set, however, the situation is different: The XL data set outperforms the classic data set in every single category (see table 4.3); in table 4.3 it can, furthermore, be seen that the coverage of the ALOD XL endpoint outperforms DBpedia on every single data set except for the OAEI Synthetic data set on whole terms.¹⁵ The outperformance is significant: When looking at the whole-term coverage, the coverage on ALOD XL often scores more than twice as good as DBpedia.

The performance on the OAEI Disease and Phenotype data set is bad on all endpoints and represents a visible outlier. When examining the labels that could not be linked, one can see that this is due to extreme tail entities, like molecules represented by their structural formula in textual form such as *N-[N-[N-(N{2}-L-Arginyl-L-lysyl)-L-alpha-aspartyl]-L-valyl]-L-tyrosine*.

It can be concluded that the ALOD data set not only contains more resources but

¹²In order to remove stopwords, the publicly available corpus by the Information Retrieval Research Group of Glasgow University was used, see http://ir.dcs.gla.ac.uk/resources/linguistic_utils/stop_words.

¹³Note that this is a very strict criterion, meaning that a term consisting of 3 tokens (e.g. *Super Galactic Empire*) is counted as not found when only two terms could be linked to a concept in the data set. In fact, terms consisting of up to 5 tokens have to be retrieved completely in order to count as a positive match. A threshold of 50% would probably lead to a significantly higher token coverage as most terms consist of few tokens.

¹⁴see http://webisa.webdatacommons.org/concept/european_union_

¹⁵Note that the OAEI Synthetic data set seems to be created of Wikipedia resources, so this is very likely a biased comparison.

Dataset		DBpedia		WebIsALOD	
Name	# of Terms (English)	Whole Term	Tokens	Whole Term	Tokens
Anatomy	11,928	0.17010	0.85680	<u>0.48625</u>	<u>0.91708</u>
Conference	488	0.18647	0.85450	<u>0.40779</u>	<u>0.96721</u>
BioMed	408,483	0.10749	0.77326	<u>0.19914</u>	<u>0.87149</u>
Disease and Phenotype	315,186	0.09125	0.37975	<u>0.16220</u>	<u>0.52120</u>
University Admission	173	0.10982	0.83236	<u>0.27167</u>	<u>0.97109</u>
Birth Registration	232	0.36206	0.89655	<u>0.48707</u>	<u>0.99137</u>
Synthetic	803	<u>0.80946</u>	0.94396	0.30511	<u>0.97883</u>
Doremus	1782	0.31481	0.58641	<u>0.35746</u>	<u>0.71212</u>
FSDM	2015	0.07806	0.90645	<u>0.31548</u>	<u>0.98710</u>

Table 4.2: Coverage Statistics DBpedia vs. ALOD XL

The best results on the whole term and on individual tokens are underlined.

can also achieve a higher coverage on those when using well known benchmark ontologies and a real data model from the financial domain.

4.4 Semantic Experiments

In order to evaluate the quality of the semantic knowledge contained in the ALOD data set and to find good configurations for features in the matching process, semantic experiments have been conducted using three different gold standards.

The data sets for the experiments presented here are publicly available and are described in detail in subsection 4.4.1. The structure and context of the creation of these evaluation data sets for concept semantics is generally the same: Annotators are presented with two terms and have to annotate the similarity/relatedness. They do this for multiple word-pairs. An example of instructions for annotators is given in figure C.6 in the appendix. Eventually, the annotated scores are averaged. [53, p. 123] [77, pp. 675-676] [22, pp. 139-140] In the end, the gold standards consist of word pairs which have an associated score.

The features developed in this thesis are evaluated by the degree of correlation they have with the gold standards. The results are reported in subsection 4.4.2.

Dataset	Improvement of ALOD XL over ALOD Classic	
	Δ Whole Term	Δ Tokens
Name		
Anatomy	+ 0.28471	+ 0.29376
Conference	+ 0.14140	+ 0.08812
BioMed	+ 0.11940	+ 0.25863
Disease and Phenotype	+ 0.13505	+ 0.46905
University Admission	+ 0.15607	+ 0.09826
Birth Registration	+ 0.21552	+ 0.06034
Synthetic	+ 0.16937	+ 0.26402
Doremus	+ 0.25870	+ 0.42986
FSDM	+ 0.14774	+ 0.03291

Table 4.3: Absolute Coverage Improvements of ALOD XL over ALOD Classic
Relative Coverages

4.4.1 Gold Standards Used

WordSimilarity-353 (WS-353) WS-353 [53] belongs to the "most commonly-used evaluation gold standard[s] for semantic models" [77, p. 671]. The data set consists of 365 noun-noun pairs which are annotated with a similarity score. It is publicly available on the Web.¹⁶ Despite its popularity, WS-353 is criticized for not clearly distinguishing between similarity and association [77, p. 671]. The inter-annotator agreement is $\rho = 0.61$ according to Hill et al. [77, p. 676].

MEN MEN is a gold standard by Elia Bruni et al. [22] consisting of 4,000 word pairs (of which 2,005 are noun-noun pairs) with a semantic relatedness rating. The data set is publicly available.¹⁷ Hill et al. report an inter-annotator agreement of $\rho = 0.68$ but also note that the actual agreement "may be somewhat lower" [77, p. 676] due to the small sample size used.

SimLex-999 SimLex-999 [77] is a gold standard which "explicitly quantifies *similarity* rather than *association* or *relatedness*" [77, p. 1]. The gold standard was created by asking 500 English speakers to rate similarity of the given word pairs (rather than association). The gold standard consists of 666 noun pairs, 111 adverb pairs and 222 verb pairs with different levels of concreteness [77, p. 666]. According to the authors, the inter-annotator agreement is $\rho = 0.67$ [77, p. 11].

¹⁶see <http://www.cs.technion.ac.il/~gabr/resources/data/wordsim353/>

¹⁷see <http://clic.cimec.unitn.it/~elia.bruni/MEN>

SimLex-999 is publicly available¹⁸ and commonly used to evaluate distributional semantic models.

4.4.2 Results

To evaluate how well a model can detect similarity and relatedness, Spearman’s rank correlation coefficient (*Spearman’s rho*) is used [77, p. 15] [22, p. 139] which is based on the rank of two variables rather than their discrete value [6, p. 35]. The formula is given in equation C.10 in the appendix.

Correlations for the features *BroaderOverlap* and *NarrowerOverlap* have been calculated for different limits l (i.e., the narrower/broader overlap of the top l concepts). Limits $l \in \{50, 100, 500, 1,000\}$ have been evaluated for the ALOD XL as well as the ALOD Classic endpoint. Jaccard is used to obtain a numeric overlap measure.¹⁹ Tables B.2, B.3, and B.4 in the appendix contain the obtained correlation values with the WS-353, MEN, and SimLex-999 gold standards.

Initially, the *Distance in Broader Vector Space* has also been evaluated using different configurations. Experiments here, however, have quickly revealed an optimum at a very low decay factor and a fixed element base value leading to the same results as the overlap features but with much more calculation effort.²⁰ The feature was, therefore, excluded from further investigations.

ALOD2Vec has been evaluated using 100 walks per entity in ALOD Classic, including reverse walks. Four different flavors were trained: 200-dimensional SG and CBOW models as well as 500 dimensional SG and CBOW models. The feature has also been evaluated on the ALOD XLR subset for 200-dimensional embeddings. Tables B.5 B.6, and B.7 in the appendix contain the obtained correlation values with the WS-353, MEN, and SimLex-999 gold standards.

All remaining features did not lead to any meaningful result when evaluated individually and are not further discussed in the following.

Table 4.4 gives an overview of the best performing correlation values for each feature group per data set.

¹⁸see <https://www.cl.cam.ac.uk/~fh295/simlex.html>

¹⁹Early experiments showed no significant difference between the Jaccard coefficient and the Dice coefficient. In order to keep the configuration space small, Jaccard has been chosen for all further experiments.

²⁰A low decay factor gives more weight to the overlap one hop away from the original concept. Good results were achieved with a factor around 0.075 which means that concepts more than one hop away contribute not even 10% to the final result. For details of this feature and its calculation refer to subsection 3.3.1 and the example in appendix C.4.

Dataset	Best Overlap	Best ALOD2Vec
WS-353	0.5376	<u>0.6599</u>
MEN	0.6826	<u>0.7202</u>
SimLex-999	<u>0.3890</u>	0.3354

Table 4.4: Best Spearman’s ρ Values for Overlap and ALOD2Vec
 The best value is underlined. Note that the best values might have been obtained by different configurations. For all correlation values see appendices B.2 and B.3.

The results clearly indicate that there is semantic knowledge contained in the ALOD data set. Concerning the overlap-based features, for all three data sets the XL endpoint performed best. Overall, the best configuration here is the top 500 narrower overlap.

ALOD2Vec performs similarly well. The optimum was always achieved using a 200-dimensional vector. Overall, the best configuration here is CBOV for ALOD Classic and SG for ALOD XLR. Nonetheless, it is notable that the differences between the various flavors are not very large. The XLR embeddings outperform the classic embeddings on all data sets except for SimLex. Generally, ALOD2Vec performs better than the overlap features on all similarity data sets when using Pearson’s ρ as benchmark rather than Spearman’s ρ .

As the gold standards are publicly available, the data can be compared to other approaches. The benchmark numbers presented in the following are taken from Hill et al.’s publication [77]. The authors provide figures for five different language models: (1) Collobert’s and Weston’s model [33], (2) Huang et al.’s model [82], (3) the Vector Space Model (VSM) [96] by Kiela et al., (4) Latent Semantic Analysis (LSA) [103], and (5) word2vec by Mikolov et al. [117, 118]. Models (1) and (2) were trained on a 150 million words corpus; models (3) and (4) were trained on a $\sim 1,000$ million words corpus. Word2Vec was trained on both. [77, p. 680]

WordSim For the WordSim data set, the best overlap configuration beats the other benchmark models LSA, VSM, and Mikolov et al. on the 150 million token corpus and also Collobert and Weston. Word2Vec trained on a 1,000 million token corpus as well as Huang et al.’s approach, on the other hand, outperform the overlap feature on the WordSim gold standard.

The best ALOD2Vec configuration (XLR SG 200) outperforms all other approaches in the benchmark group with $\rho_{Spearman} = 0.6599$ close to the runner-up which is the word2vec model trained on a 1,000 million corpus with $\rho_{Spearman} = 0.655$. Figure 4.5 shows the relative performance of the best ALOD feature on the gold standard. Detailed numbers are given in the appendix in tables B.2 and B.5.

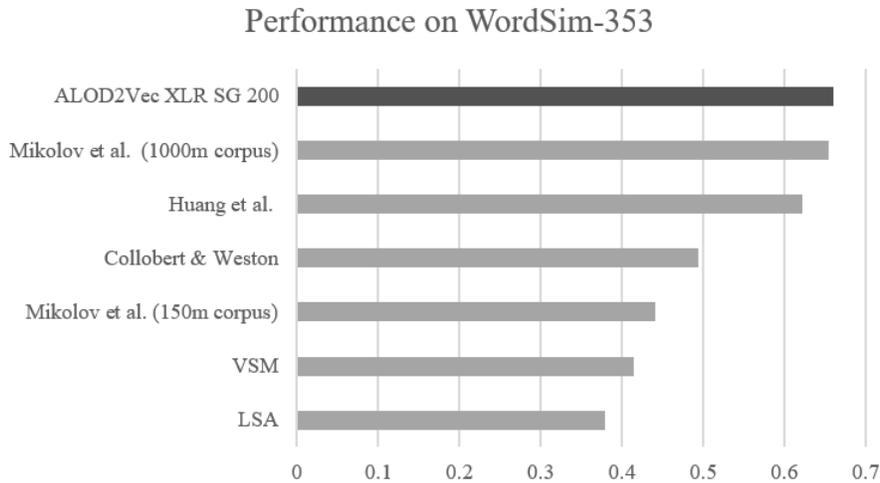


Figure 4.5: Performance on WordSim

MEN On the MEN gold standard, the top 500 narrower overlap scores second place compared to the other approaches with $\rho_{Spearman} = 0.6826$. Word2Vec trained on the larger corpus slightly outperforms the approach with $\rho_{Spearman} = 0.699$.

The best ALOD2Vec configuration (XLR SG 200) beats again all other approaches in the benchmark with $\rho_{Spearman} = 0.7202$.

It has to be noted that only the nouns of the MEN standard were used to calculate the correlation for the ALOD method. Figure 4.6 shows the relative performance of the best ALOD feature on the gold standard. Detailed numbers are given in tables B.3 and B.6 in the appendix.

SimLex-999 On the SimLex gold standard, the features also perform quite well: The overall top configuration (narrower overlap of the top 500 concepts on the XL endpoint, $\rho_{Spearman} = 0.3890$) outperforms all approaches in the benchmark except for Mikolov’s word2vec when trained on a 1,000 million token corpus ($\rho_{Spearman} = 0.414$). This is also the case for all configurations of the ALOD2Vec feature.

It has to be noted, though, that the comparison is not fully accurate as in the work presented here, only two thirds of the data set (666 nouns) were used whereas the other models were benchmarked on the whole set. Figure 4.7 shows the relative performance of the best ALOD feature on the gold standard. Detailed numbers are given in the appendix in tables B.4 and B.7.

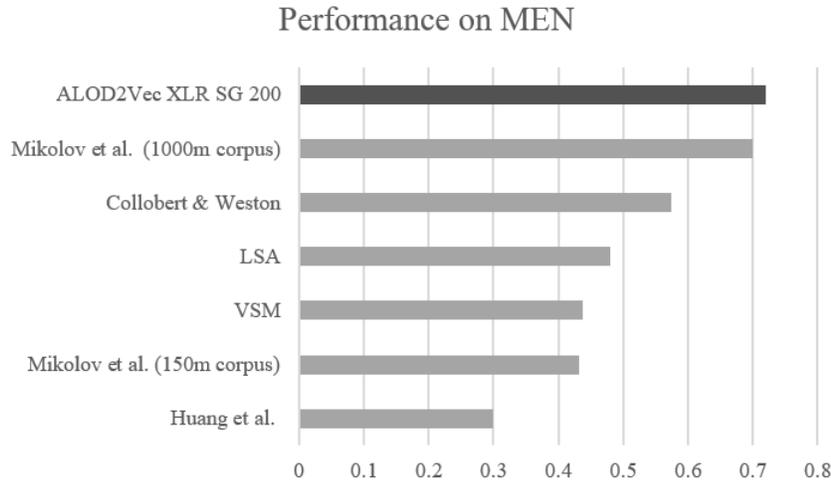


Figure 4.6: Performance on MEN

It can be seen that the task of similarity score calculation is very hard compared to relatedness score calculation: All models achieve lower correlation scores on this data set compared to the others where the focus is rather on relatedness.

4.5 Regressions on Gold Standards

4.5.1 Feature Configuration

In order to find out whether the features can be combined, regressions were performed on the MSGS gold standard. The following five features were used in regressions:

1. Broader Concept Overlap (top 1,000)
2. Narrower Concept Overlap (top 500)
3. One has Other as Broader Concept
4. Number of Narrower Concepts (top 500, level 1)
5. Number of Broader Concepts

For a description of the features, see section 3.3. When using the feature *One-HasOtherAsBroaderConcept* on the XL data set with level 2, every single word

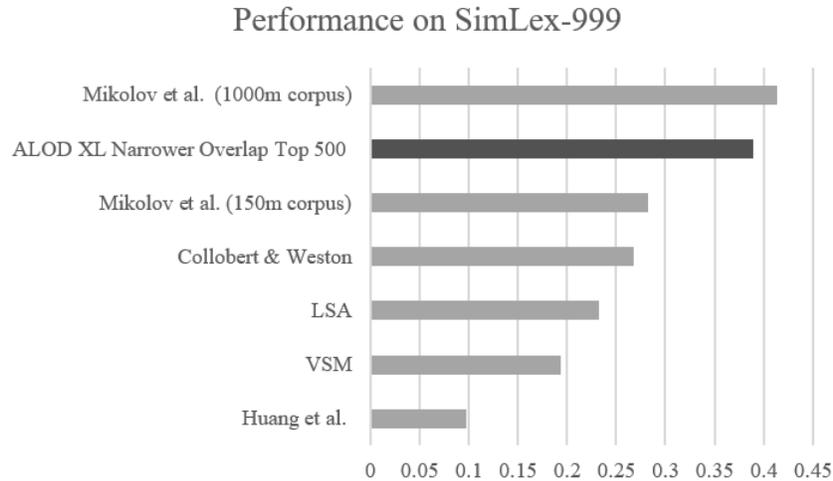


Figure 4.7: Performance on SimLex

pair of the MEN gold standard yields 1. This is a strong indicator for the noise in the XL data set. In order to use the feature, the level was set to 1 and the overlap was restricted to the top 500 concepts. For the broader and narrower Jaccard overlap, the best configurations of the previous experiments have been used, i.e., top 500 narrower and top 1,000 broader concepts. Features were calculated for the ALOD Classic as well as the ALOD XL data set.

A program was implemented in Java which accepts a gold standard in CSV format as well as features of class `FeatureGenerator` and outputs a file with the numerical features in CSV format. The latter was then used to run the regressions in RapidMiner. The implementation is generic, i.e., not restricted to the features presented here.

4.5.2 Results

The MSGS-1234 gold standard (see 3.5.2) was used for feature aggregation. Two regressions were performed: (i) A regular linear ridge regression as well as (ii) a *least absolute shrinkage and selection operator (Lasso)* regression. Regression (i) was configured with *M5 prime* feature selection and a ridge parameter of $1 \cdot 0E-8$. The Lasso regression (ii) was additionally used because it is more aggressive than the regular linear regression in the sense that it sets coefficients to zero whereas the regular linear regression will only result in very small coefficients [54, pp. 4-5]. It can, therefore, also be used for feature selection (see [54]), i.e., the reduction of

features in order to avoid overfitting, improve interpretability, and allow for better interpretation. Results are presented in table 4.5.

	RMSE	Absolute Error	w_1	w_2	w_3	w_4	w_5	Y
Lasso XL	0.446 +/- 0.021	0.4 +/-0.018	0	0	0	1.087	0.374	0.274
Regular XL	0.43 +/- 0.025	0.376 +/-0.022	-4.504	2.75	-0.099	0.966	1.1541	0.372
Lasso Classic	0.451 +/- 0.029	0.407 +/- 0.024	0	0	0	0.32	0.231	0.271
Regular Classic	0.427 +/- 0.033	0.364 +/-0.03	-0.867	0	-0.221	0.441	0.841	0.418

Table 4.5: MSGS-1234 Regression Results

RMSE refers to *Root Mean Squared Error*, w_i refers to the assigned weights where i describes the feature as numbered in 4.5.1; Y refers to the intercept with the y-axis.

Interpretation One can quickly see that the regression results are of low quality given a root mean squared error in the 0.4 range on a binary label. This could be confirmed using tests on the OAEI Anatomy data set. The results are, therefore, discarded and not used in the following.

The regression outcome allows for multiple interpretations: First, one could argue that the given features are not fit for the task. Another argument might be that the gold standard is not large enough and leads to overfitting.²¹ It might also be the case that the samples used are not meaningful. A combination of all those reasons is possible as well. Given the good semantic results presented in 4.4, one can most likely reason that the gold standard is not fit for the task at hand.

Because of the bad results obtained with regressions, the approach was not further explored.

4.6 Ontology Matching Experiments

In this section, the implemented matcher is evaluated in various configurations against four gold standards which are introduced in the following subsection. The

²¹The results are in line with earlier experiments on a smaller MSGS gold standard. MSGS was extended as a consequence but without a positive effect. With 1234 annotations, however, the gold standard is still small.

results are presented in 4.6.3. The best configuration is submitted to the OAEI 2018 campaign. Expected results there are presented in 4.6.4.

4.6.1 Gold Standards Used

OAEI Anatomy In the OAEI Anatomy track, two large ontologies have to be matched: (1) The *Adult Mouse Anatomy* and (2) a subset of the *National Cancer Institute (NCI) Thesaurus*²² which describes the human anatomy. The vocabulary used is domain-specific (biology domain). Reference alignments as well as the data set itself are available online.²³ The Adult Mouse Ontology has 2,744 classes and the NCI Thesaurus has 3,304 classes. The amount of properties in this track is negligible.

OAEI Conference The OAEI Conference track is composed of 16 ontologies from the domain of conference organization. All ontologies are taken from the *OntoFarm* collection [179] and are rather small (see table 4.6). The motivation behind the collection was to find ontologies that are heterogeneously structured but still describe the same domain [179, p. 46]. This setting makes the collection suitable for ontology matching. The semantic information within the ontologies is rare and often only embedded within the IRI. An example would be `http://cocus#Event_Setup`; in this case, no label or description is given and the most valuable semantic information has to be extracted from the IRI. There are three reference alignments:

1. Original Reference Alignments (*ra1*)
They are publicly available for download.²⁴
2. Entailed Reference Alignments (*ra2*)
They are a transitive closure computed on *ra1* where conflicting correspondences are manually resolved. Therefore, the correctness and completeness of *ra2* is expected to be better than that of *ra1*.
3. Violation Free *ra2* (*rar2*)
For this evaluation data set, violating correspondences were automatically identified using [162, 161]²⁵ and then manually resolved.

²²see <https://ncit.nci.nih.gov/>

²³see <http://oaei.ontologymatching.org/2017.5/anatomy/index.html>

²⁴see <http://oaei.ontologymatching.org/2017/conference/eval.html>

²⁵The underlying idea of the violation detection is that mappings should not lead to new semantic relationships between concepts of one (input) ontology which is also known as *conservativity principle* [161, p. 2].

Name	cl	dp	op	Related Link
Ekaw	74	0	33	http://ekaw.vse.cz
Sigkdd	49	11	17	http://www.acm.org/sigs/sigkdd/kdd2006
Iasted	140	3	38	http://iasted.com/conferences/2005/cancun/ms.htm
Cmt	36	10	49	http://msrcmt.research.microsoft.com/cmt
Edas	104	20	30	http://edas.info/
Conference	60	18	46	-
ConfOf	39	23	13	-

Table 4.6: OAEI Conference Track Statistics

The table is a composition of one given at the OAEI Web page²⁷ and one in the original paper of the data set [179, p. 49]. *cl*, *dp*, and *op* refer to the number of classes, data properties, and object properties in the corresponding ontology.

Even though the OAEI track uses all three alignments, only a part of *ral* is publicly available for download. Table 4.6 gives an overview of the available ontologies together with statistics about the data set.

OAEI Large BioMed The *OAEI Large Biomed Track* consists of three very large ontologies: Foundational Model of Anatomy (FMA)²⁸, National Cancer Institute Thesaurus (NCI)²⁹, and SNOMED CT³⁰. The reference alignments are based on the Unified Medical Language System (UMLS) Metathesaurus³¹. The biggest challenge of this particular track is the sheer size of the ontologies. In addition to the whole ontologies, subsets of the ontologies are also provided as an additional task allowing evaluation also for matchers that cannot handle the full size of the data sets. Table 4.6.1 gives an overview of the number of classes in each ontology.

²⁷see <http://oaei.ontologymatching.org/2017/conference/index.html>

²⁸see <http://si.washington.edu/projects/fma>

²⁹see <https://ncit.nci.nih.gov/ncitbrowser/>

³⁰see <https://www.snomed.org/>

³¹see <https://www.nlm.nih.gov/research/umls/>

Ontology	All Classes	Small Overlap NCI	Small Overlap FMA	Small Overlap SNOMED
NCI	66,724	-	6,488	23,958
FMA	78,989	3,696	-	10,157
SNOMED	122,464 ³²	51,128	13,412	-

Table 4.7: OAEI Large BioMed Statistics

The numbers refer to the number of classes in the corresponding ontology. They are compiled from the downloadable material of the track³³.

SAP Financial Services Data Management Data Model As mentioned in the introduction, the topic of ontology matching is also of interest to businesses. The aforementioned FSDM data model was provided by SAP SE for the purpose of alignment evaluation. Currently, most FSDM mappings are under development; however, one is already completed and was provided to the author of this thesis. The official product consists of two data model parts: *A conceptual data model (CDM)* and a concrete implementation for the SAP HANA database in the form of a *physical data model (PDM)*. The data models as well as the mapping between them are available in SAP PowerDesigner³⁴, SAP's proprietary data modeling tool. An interesting aspect is that all information in the CDM is also represented in the PDM. Despite this fact, the mappings are not as trivial as one might think: In the CDM, there are 400 entities; in the PDM, there are 230 entities.³⁵ This is due to a performance-optimized data model which makes heavy use of data model denormalization and table merging. Tables in the PDM tend to have rather general names as they comprise a lot of specializations. The main difference between the data models is their level of granularity.

In order to apply an ontology matching algorithm, both data models have to be translated into an ontology and the mappings between them have to be extracted. Therefore, a PowerDesigner extension was implemented in Visual Basic Script

³²Note that for the SNOMED ontology this number does still not represent the whole ontology but the subset with NCI and FMA. In the track, however, the full ontology is not provided.

³³see http://www.cs.ox.ac.uk/isg/projects/SEALS/oaei/2017.5/LargeBio_dataset_oaei2017.zip

³⁴see <https://www.sap.com/products/powerdesigner-data-modeling-tools.html>

³⁵Technical tables in the PDM which carry no semantic meaning are ignored (also not present in the ontology). All numbers are valid as of Release 1.1.

(VBS).³⁶ Static rules were defined to translate the data models into OWL ontologies. They can be found in appendix C.9. The mappings are extracted in the XML format given by the Alignment API.

4.6.2 Features Evaluated

The *broader vector space* feature showed good semantic results but turned out to be too expensive due to the exponential amount of queries to be asked. Furthermore, semantic experiments showed that the optimum lies at a very low decay factor (between 0.05 and 0.1 on all three data sets). However, at such a low decay factor, the approach is equal to a simple overlap because the most weight (> 90%) is given to the overlap at the first hop. Therefore, this feature was discarded for concrete matching tasks.

Broader overlap and *narrower overlap* are evaluated in actual matching tasks using their best configurations in semantic experiments, i.e., top 1,000 broader overlap and top 500 narrower overlap on the XL endpoint.

All RDF2Vec feature configurations lead to good semantic results and are also evaluated in the following.

The remaining features did not perform well in the semantic experiments; the underperformance could be confirmed in early ontology matching tests. Thus, they are not further evaluated here.³⁷

In addition, experiments showed that the matcher profits from using a matching restriction strategy rather than a plain strategy (see subsection 3.6.1); in this case though, the differences between a one-to-many and a one-to-one strategy were negligible. Therefore, results are reported in the following for the implemented one-to-many strategy in order to reduce the configuration space.

4.6.3 Matcher Results

OAEI Anatomy Data Set

The matcher results for the Anatomy data set are given in table 4.8. The overlap-based matchers perform slightly better than the given baseline by the OAEI³⁸ with

³⁶A PowerDesigner extension is similar to a Microsoft Excel macro: It allows to implement additional functionality and provides access to data structures that are internally used by the application [150, p. 9].

³⁷One exception are the implemented string-based methods which do not yield good semantic results but show mediocre performance in the matching task. However, as it makes not much sense evaluating those on their own for the scope of this thesis, they are discarded as well.

³⁸see <http://oaei.ontologymatching.org/2017/results/anatomy/index.html>

the broader overlap performing better than the narrower overlap. All ALOD2Vec matcher configurations outperform the overlap-based ones. The overall best matcher configuration is the ALOD2Vec Classic CBOW 500.

The overall optimal thresholds show that the data set is driven by lexical matches. When evaluating the false positives, typical candidates are homonyms or labels that share a very large part of the tokens but are different. However, the precision is very high for all matchers. Typical false negatives are close lexical matches which contain adjectives such as (*inner ear*, *internal ear*, =). The true positives are mostly exact lexical matches or share many common tokens. Examples for some true positives are given in table 4.9.

Matcher	F_1	Precision	Recall	Best Threshold
Narrower Overlap 500 XL	0.7818	0.9919	0.6451	0.85
Broader Overlap 1,000 XL	0.7830	0.9959	0.6451	0.8
ALOD2Vec Classic CBOW 200	0.7851	0.9949	0.6484	0.9
ALOD2Vec Classic CBOW 500	<u>0.7861</u>	0.9949	0.6497	0.85
ALOD2Vec Classic SG 200	0.7845	0.9959	0.6471	1.0
ALOD2Vec Classic SG 500	0.7850	0.9959	0.9478	1.0
ALOD2Vec XLR CBOW 200	0.7845	0.9929	0.6497	0.85
ALOD2Vec XLR SG 200	0.7851	0.9949	0.6481	0.9
OAEI Baseline ³⁹	0.766	0.997	0.622	-

Table 4.8: OAEI Anatomy Matching Results

The best matcher of each category is printed in bold. The overall best matcher according to F_1 is additionally underlined.

	True Positives	
Broader 1,000	superior cerebellar vein	superior cerebellar cistern
	pupillary membrane	membrane
	capillary	capillary
	middle caudal artery	middle hemorrhoidal artery
	trochlear IV nerve	trochlear nerve
CBOW-500	eye chamber	chamber of the eye
	forebrain	fore-brain
	skin fluid/secretion	skin fluid or secretion
	head/neck muscle	head and neck muscle
	liver right lobe	right lobe of the liver

Table 4.9: Examples for True Positives on the Anatomy Data Set

OAEI Conference Data Set

The OAEI Conference data set is benchmarked using micro average. As just one reference data set is publicly available for download, only the `ra1` gold standard (see 4.6.1) could be evaluated. The overlap-based matchers as well as the RDF2Vec-Classic-based ones perform equally well. The best results were achieved with ALOD2Vec XLR SG 200 resulting in $F_1 = 0.5873$. Table 4.6.3 comprises the matcher results. The differences among the different configurations are marginal.

Matcher	F_1	Precision	Recall	Best Threshold
Overlap (all)	0.5841	0.7123	0.4951	0.86
ALOD2Vec Classic (all)	0.5841	0.7123	0.4951	0.86
ALOD2Vec XLR CBOW 200	0.5869	0.7136	0.4983	0.82
ALOD2Vec XLR SG 200	<u>0.5873</u>	0.7083	0.5016	0.78
OAEI Baseline ⁴⁰	0.56	0.8	0.43	-

Table 4.10: OAEI Conference Matcher Results

In this table, the micro averages are given. The best score is bold-printed and underlined.

OAEI Large BioMed Data Set

Due to the sheer size of the large BioMed data sets, the FMA-NCI subset was chosen for evaluation of this track. The overlap-based measures did not finish within a given time-frame of 8 hours. However, the ALOD2Vec-based approaches ran under 3 hours and the results are reported in the following. The campaign does not have a string-distance baseline, but rather the average is given. Note that this is a skewed baseline as it contains some matchers that use the UMLS-Metathesaurus as background knowledge (such as XMAP [41]) which leads to an advantage because the reference alignments are based on the very same metathesaurus. In addition to the average, the results of a similar approach are, therefore, also given in the following – namely WikiMatch in its latest form *WikiV3* [72]. Table 4.11 comprises the results. As before, the ALOD2Vec configurations perform similarly well with CBOW 500 on ALOD Classic achieving the highest F_1 of 0.8875. Differences among the different configurations are minor.

⁴⁰see <http://oaei.ontologymatching.org/2017/conference/eval.html>

Matcher	F_1	Precision	Recall	Best Threshold
ALOD2Vec Classic CBOW 200	0.8872	0.9743	0.8145	0.9
ALOD2Vec Classic CBOW 500	<u>0.8875</u>	0.9739	0.8151	0.85
ALOD2Vec Classic SG 200	0.8874	0.9728	0.8158	0.9
ALOD2Vec Classic SG 500	0.8873	0.9724	0.8158	0.9
ALOD2Vec XLR (all)	0.8870	0.9743	0.8142	0.9
OAEI 2017 Average ⁴¹	0.891	0.946	0.844	-
WikiMatch V3 ⁴²	0.797	0.883	0.726	-

Table 4.11: OAEI LargeBio Matcher Results

The best matcher of each category is printed in bold. The overall best matcher according to F_1 is additionally underlined.

SAP Financial Services Data Management Data Model

As the underlying ontologies for the SAP FSDM business use case were created within the scope of this thesis, there is no benchmark for this particular alignment task. Therefore, the matching was evaluated also on another approach, namely LogMap [87]. The approach was chosen as it is consistently one of the best matching approaches in recent OAEI campaigns (see [88], [89], or [90]). The approach was evaluated on all OAEI tracks and, therefore, is likely not too restricted to one data set but rather a general-purpose approach. The coding is publicly available.⁴³ The results can be found in table 4.12.

⁴⁰see <http://www.cs.ox.ac.uk/isg/projects/SEALS/oei/2017/results/>

⁴¹see <http://www.cs.ox.ac.uk/isg/projects/SEALS/oei/2017/results/>

⁴³The LogMap source code is publicly available on GitHub: <https://github.com/ernestojimenezruiz/logmap-matcher>. A packaged stand-alone application in the form of a jar is also available for download: <https://sourceforge.net/projects/logmap-matcher/files/Standalone%20distribution/>.

Matcher	F_1	Precision	Recall	Best Threshold
Narrower Overlap 500 XL	0.7485	0.8551	0.6655	0.55
Broader Overlap 1,000 XL	0.7485	0.8551	0.6655	0.55
ALOD2Vec Classic CBOW 200	0.7485	0.8551	0.6655	0.55
ALOD2Vec Classic CBOW 500	0.7469	0.8512	0.6655	0.55
ALOD2Vec Classic SG 200	0.7485	0.8551	0.6655	0.6
ALOD2Vec Classic SG 500	0.7485	0.8551	0.6655	0.6
ALOD2Vec XLR CBOW 200	0.7469	0.8512	0.6655	0.7
ALOD2Vec XLR SG 200	0.7485	0.8551	0.6655	0.7
LogMap	0.7459	0.8545	0.6618	0.4

Table 4.12: SAP FSDM Matcher Results

The optimal F_1 score is 0.7485 and is achieved by multiple configurations.

With an F-Score of ≈ 0.75 , LogMap performs quite good and competitive compared to the matchers of this thesis. It can be seen that the matcher is also strong in a real enterprise setting. To the knowledge of the author of the thesis, this is the first time that the matcher has been evaluated in such an environment.

The matchers of this thesis perform almost all equally and only slightly better than LogMap. This is the only data set where all matchers presented in this thesis outperform LogMap.

4.6.4 OAEI Participation

The matcher using CBOW 200 embeddings is also registered for the OAEI Campaign 2018. This configuration has been chosen because of its good performance compared to the other configurations evaluated in this thesis and because of a relatively good runtime that allows participation in the LargeBio track. In addition, given similar outcomes, a 200-dimensional embedding is to be preferred because of less storage requirements and higher performance.

Results are expected to be published later this year (2018). Yet, by evaluating the submitted SEALS package locally, some performance figures can already be anticipated. They are comprised in table 4.13.⁴⁴ Note that the final numbers will be more comprehensive as not all gold standards are publicly available. The fixed threshold for the submission is $t = 0.9$.

⁴⁴While for the figures reported before a best threshold was determined, the matcher registered uses only one threshold. Therefore, the results differ from the ones presented before.

	F₁	Precision	Recall
Anatomy	0.785	0.995	0.648
Conference	0.584	0.712	0.495
Large BioMed⁴⁵	0.901	0.972	0.839

Table 4.13: Expected Matcher Performance OAEI 2018

Given these figures, the matcher outperforms WikiMatch-V3 on the Conference data set ($F_1 = 0.57$) as well as on the Large BioMed data set ($F_1 = 0.79$). On the Anatomy data set, WikiMatch outperforms the registered matcher with $F_1 = 0.802$.

4.6.5 Results Summary

The ALOD-based features add value to the matching process and performed all stronger than the string-distance baseline. Generally, it can be seen that ALOD can be used for ontology alignment. However, the results are often only some percentage points better than the string-distance baseline, the improvement is not large.

All matcher configurations performed similarly well. In the case of the ALOD2Vec approach, there was no clear performance gain in using 500 dimensional vectors and the improvement in using the larger XLR data set was also relatively low. The overlap-based configurations fall short compared to the ALOD2Vec ones when it comes to evaluation results as well as runtime performance.

The ALOD2Vec Classic CBOW 200 matcher will also be evaluated in the OAEI 2018 campaign. It is expected that the performance is similar to other Web-based matchers.

Furthermore, it could be shown that LogMap is not restricted to the OAEI data sets but performs also strong in a real-world business scenario.

4.7 SAP FSDP Matching Use Case

4.7.1 Automatic Schema Matching at SAP

Despite good results on the corporate data set presented, fully automated matching is still not precise enough for an enterprise setting where each and every corres-

⁴⁵Note that the evaluation here ignores flagged repairs while the performance figures on the data set reported before do not. This explains the slight difference in the numbers even though the threshold is the same.

pondence must be correct.

In addition, another key requirement that was identified in interviews at the company is the option to define rules for each correspondence. Up to now, there is no matcher that can achieve this automatically.

Nonetheless, the research of this thesis is relevant in a business setting. The next section presents a concrete prototype for using the algorithms of this thesis for supporting the matching process.

4.7.2 FSDM Semantic Search

In interviews with people actively involved with the task of creating mappings for FSDM, one particular pain point that was mentioned several times was the issue of finding concrete entities without knowing their name. A consultant, for instance, might want to find the relevant entity used for *customer*. As PowerDesigner only performs a boolean search, looking for the concept returns no results. This is due to the fact that the actual entity is called *BusinessPartner* in FSDM. An ontology matcher will only return one result, i.e., the matching; the user in this case, however, is interested in the top n related concepts. Given that multiple similarity calculators were already implemented, a quick solution in the form of a simple prototype was implemented as Web server. A consultant can connect to the server in his terminal and retrieve the top n closest semantic FSDM concepts to her search term. This can help her not only for schema matching but also when building views. An exemplary search is depicted in figure 4.8. A quick user guide on how to use the server can also be found in appendix C.10. The server is implemented in Java and any subclass of `FeatureGenerator` can be used as search algorithm, i.e., all features presented in this thesis.

```

.....
...SSS.....AAAA.....PPPPPP.....
..SS..SS.....AA..AA.....PP...PP...
..SS.....AA...AA.....PP...PP...
...SS.....AAA...AAA.....PPPPPP...
...SS.....AA..AAA..AA...PP.....
..SS..SS..AA.....AA..PP.....
...SSS...AA.....AA..PP.....
.....
.....
-----
Main Menu
-----
What do you want to do?
1) Find an Entity
c) Configuration
x) EXIT
Your Input: 1
-----
SAP FSDM Semantic Search
-----
Search Term: business partner
Your Input: business partner
Associated Concepts:
http://webisa.webdatacommons.org/concept/business_partner_
Please hold the line. Search in progress...

The following FSDM entities were found:
1) BusinessPartner
2) Company
3) Organization
4) ConsolidatedBusinessPartner
5) IndividualPerson
-----
Main Menu
-----
What do you want to do?
1) Find an Entity
c) Configuration
x) EXIT
Your Input:

```

Figure 4.8: FSDM Semantic Search: Depicted is an exemplary search process. After the user connected to the server she is prompted to enter what she would like to do. She chooses to search for "business partner" and the top 5 results are returned. Note in this case that *Company*, *Organization*, and *IndividualPerson* are indeed relevant results as they are specializations of *BusinessPartner*.

4.7.3 Business Value of this Thesis

In addition to the search prototype presented, the *PowerDesigner Ontology and Mapping Extractor Extension*, and the newly created *FSDM Ontology*, a meta structure was developed for the company so that future mappings and the rules between them can be persisted in a uniform format. This allows the department to collect mapping data in a machine-readable way and to use this data for further research later on.

Moreover, two inventions were made in the context of this thesis: By proving that the financial data model can be viewed as a graph, direct implications for SQL view building were recognized. Two patents were filed out of which one is already registered in the US [19]. Note that everything presented in this thesis is in the public domain. The inventions patented are loosely related to the topic at best but are a valuable by-product of this thesis for SAP.

Chapter 5

Conclusion

5.1 Critical Remarks

5.1.1 Data Sets

OAEI Anatomy Data Set Concerning the anatomy data set it has to be mentioned that the OAEI gold standard alignments have a heavy exposure to lexical similarity which can already be seen by the baseline string equivalence solution of 0.766¹. This is an advantage for algorithms exploiting the lexical structure, like LogMap [87] which achieves a very high f-measure without much effort.

FSDM Data Set The FSDM data set was created within the scope of this thesis. It is a real integration scenario that is actually used in business. However, out of the multitude of integration scenarios, the selected scope can be considered to be a rather simple task: First, the data models are from the same vendor – i.e., SAP SE – which already might be an indication for the semantic heterogeneity being smaller compared to situations where models of different vendors are to be matched. Second, the data model is rather small as opposed to very mature operational data models in the banking sector. Lastly, the lexical overlap is still relatively high.

Remaining Data Sets Interestingly, the textual overlap on all data sets evaluated is high. However, due to the insufficient data situation it cannot be concluded whether this is the norm or an anomaly.

¹see <http://oaei.ontologymatching.org/2016/results/anatomy/index.html>

5.1.2 Semantic Experiments

Despite very good results in semantic experiments, the matcher does not outperform current top-notch OAEI approaches. One important aspect about the data sets has to be highlighted here: The concepts used in the semantic experiments are rather common ones like *river, flower, sun* (MEN), *winter, child, book* (SimLex) or *paper, plane, football* (WordSim). In matching tasks, on the other hand, the vocabulary is much more domain specific. As a consequence, very good results on semantic experiments can still lead to unsatisfactory results in the matching task due to the vocabulary mismatch.

Furthermore, the semantic experiments were restricted to nouns but the matching data sets were not.

5.1.3 Ontology Matching and Evaluation Tools

Even though the syntactic heterogeneity is very low due to common standards for ontologies and reference alignments, and even though the OAEI organizes campaigns since 14 years as of now, the provided tool set for ontology alignment development and evaluation is not satisfactory. The Alignment API is important for defining a common interface and allows for simplistic evaluation of a matcher; however, only slightly more complex tasks such as calculating micro and macro averages, are not supported anymore. Comprehensive functions required for the development of a matcher – like calculating the number of non-trivial correspondences found or outputting false positives and false negatives – are not possible out-of-the box.

Similarly, the SEALS framework provides rather crude evaluation functions. In addition, the framework is not intuitive and not easily usable. A significant amount of time has been spent resolving framework incompatibilities leading to a reimplementation of the persistence approach used in this thesis because the problem could not be solved even after contacting an expert.

The Hobbit platform might be the right idea: Creating a common cloud platform for matcher evaluation is certainly helpful. However, as of now the performance on the platform is quite bad² and the evaluation capacities are limited to precision, recall, and F-measure. Furthermore, the amount of boilerplate code required for the packaging and uploading a matcher to Hobbit is cumbersome and error-prone. There are approaches in simplifying the overall process³ but there still is a long

²A simple string matcher on the OAEI Anatomy data set can take well above one day until results are online.

³The `ontMatching` template by Sven Hertling is a good step in the right direction, see <https://github.com/sven-h/ontMatchingHobbit>.

way to go.

Due to the lack of tooling, a significant amount of time of this thesis was spent on developing a basic technical infrastructure. There certainly is upward potential when it comes to ontology alignment tooling.

5.2 Limits

5.2.1 ALOD Data Set

As the Web is no reviewed source of truth and since ALOD is based on the Web, the data set contains a lot of noise and should not be used as the only source for knowledge if reliable information is required. The concept of *Bill Clinton*⁴, for instance, has *George Bush* as broader concept with a relatively high confidence value of 0.7357.

Furthermore, the corpus used in this thesis is not up-to-date because it is based on an older version of the common crawl. The concept of *Donald Trump*⁵, for example, does not have any indication that he is the 45th President of the United States. Nonetheless, this issue can be easily resolved by updating the data set.

Another challenge is the fact that the information given in the data set can be very subjective and also conflicting. According to ALOD, *Donald Trump* is a *genius* as well as a *lunatic*, *racist*, and a *buffoon*.

Despite the good coverage numbers, the ALOD data set also suffers from the tail entity problem: While popular entities have many broader and narrower concepts on ALOD (*president*⁶ has 55,675), tail-entities do exist but have very few relations to other concepts. An example would be *cerebral dura mater*⁷ which has two broader concepts and no narrower ones. Another example would be *iliopsoas*⁸ and *iliopsoas muscle*⁹: The first one has 6, the latter one has 18 broader concepts. Even though describing the same real world concept, none of those is overlapping. The distribution analysis of relations revealed that most concepts are probably not usable since they barely have any relations. It can be concluded that the good coverage numbers are deceiving at first sight.

⁴see http://webisa.webdatacommons.org/concept/bill_clinton_

⁵see http://webisa.webdatacommons.org/concept/donald_trump_

⁶XL IRI: <http://webisa.webdatacommons.org/concept/president>

⁷XL IRI: <http://webisa.webdatacommons.org/concept/cerebral%20dura%20mater>

⁸XL IRI: <http://webisa.webdatacommons.org/concept/iliopsoas>

⁹XL IRI: <http://webisa.webdatacommons.org/concept/iliopsoas%20muscle>

Because of the automatic text extraction approach, the data set lacks the distinction of homonyms: *Apple*¹⁰, for instance, is a *fruit crop* (0.8608 confidence) as well as a *silicon valley company* (0.8288 confidence). As the example demonstrates, the lack of word sense disambiguation increases the noise for concepts with multiple meanings.

Moreover, the hypernymy graph is not consistent, i.e., two concepts can both be hypernyms of each other at the same time.¹¹ There are even concepts, where the concept itself is listed as a hypernym (e.g. *piece of the puzzle*¹² has itself as a hypernym on second position when ranked according to confidence).

Lastly, it has to be noted that the data set is only available in the English language which limits the amount of use cases. From a technical perspective, though, it should be possible to apply the same extraction process to other languages when the patterns are translated.

5.2.2 Linking to LOD Resources

Although linking to LOD resources works well, the approach is very expensive when it comes to the number of queries performed. This is due to the fact that after every string modification and every time after a token is removed, the data set is queried. When using the SPARQL online access point, this leads to a low performance induced by network transmission time. From an architectural perspective, it would be better to just submit the query term, do the processing on the server side and return the result – as it is done by typical search APIs. In such cases, specialized data structures can be used as SPARQL queries are likely not the best option from a performance perspective. The current approach was chosen because no such API exists for the ALOD data sets.

The implemented linker generally prefers longer concepts which are more specific. However, in some cases this might lead to the situation where a found concept is not valuable due to very few hypernymy relations as described in the paragraph above. Nonetheless, this is rather a restriction of the data set used.

Another restriction is that the linker does not find concepts with similar writing. Searching concepts using the term *Tolkien* will not return the concept *J. R. R. Tolkien* despite the high likelihood of referring to the same underlying concept. For such operations, a real search API with optimized data structures would be required.

¹⁰see http://webisa.webdatacommons.org/concept/_apple_

¹¹In fact, this is not a seldom phenomenon but very common. This can be easily proved by setting a high level when using features *A Has Broader Concept B* and *B Has Broader Concept A* (see 3.3.1) simultaneously.

¹²see http://webisa.webdatacommons.org/concept/_piece_of+the+puzzle

5.2.3 MSGS-1234

The gold standard for monosemous synonymy gold standard performed bad in the regressions performed. The indication is strong that the gold standard is not very useful because of its small size.

In addition, the concepts used are rather uncommon, like *bouquet* or *nosegay*, due to the strict monosemy restriction. Retrospectively, it can be seen that feature selection should have been performed on a larger, less restricted gold standard with more common terms.

5.2.4 ALOD Matcher

It could be observed that the addition of the ALOD data set can add value to a matcher and outperforms string-based methods. Nonetheless, the outperformance is not dramatically large. String-based methods work surprisingly well on all data sets evaluated. This observation has also been made earlier [71], the real challenge is to detect the remaining non-trivial cases.

One of the largest drawbacks of the matcher – and probably one of the main reasons for a rather mediocre performance – is the missing handling of non-nouns. Many labels and descriptions contain adjectives which cannot be linked and are, hence, regarded as an unknown increasing the penalty score.

Concerning the techniques to ontology matching, it has to be noted that the current matcher does only work with labels but completely ignores other information such as the structures of the ontologies to be aligned. This makes the matcher presented here vulnerable to homonyms. Nonetheless, it is easily possible to embed the features presented in this work in a more sophisticated and comprehensive alignment approach.

Another limitation of the matcher is the restriction to only equivalence relations (=) and the restriction to non-complex alignments, exclusively. Although this is very common for OAEI matchers, a real solution to the ontology matching problem and also to data integration for companies would require a matcher to recognize such cases and to be able to resolve those.

The matcher presented in this thesis is a *one-size-fits-it-all* solution. The idea that there is one configuration which performs well on any data set is not realistic, though. The OAEI Anatomy data set, for instance, profits from very aggressive String-based techniques, like applying a Porter stemmer, whereas the conference data set does not.

Lastly, the matcher presented in this thesis can only match English ontologies. Cross-language matching is not supported due to the monolingual data set used. Theoretically, a dictionary could be used but it is very likely that this would deteri-

orate results due to information lost in translation and the linking process.

5.3 Challenges for Web-Based Matchers

One challenge for most matchers is the infinite size of the configuration space. Even though good configuration ranges have been heuristically determined, e.g. by optimizing semantic relatedness scores, the number of configuration parameters is still very high in the implementation at hand. This likely accounts to all Web matchers as the number of ways in which Web knowledge might be utilized are infinite.

Another challenge is the sheer size of the Web and Web-based data sets such as WebIsALOD. Simple processes, like a random-walk generation on a very large data set, are very expensive and time consuming. Many graph algorithms are not optimized for Web-scale graphs. In addition, frameworks such as Apache Jena cannot be used efficiently on those large RDF graphs and the only option is to implement algorithms that operate on file level. Furthermore, the size requires sufficient hardware. The server used in this thesis with more than 120 GB of memory quickly reached its limit.

Also related to Web-based data is data quality. While there is valuable information on the Web (and consequently in the ALOD graph), the amount of noise is very challenging. A matcher must be able to judge the quality of a statement. The WebIsALOD confidence score is a good step into this direction but still not precise enough.

Concerning performance, like other Web-request-based matchers, the SPARQL-based approaches heavily suffer from transmission times and SPARQL overhead on the Web. Here, vector-based representation for concepts are more interesting as they perform well when stored locally and do not require a network connection (at the cost of higher disk/RAM requirements).

Lastly, handling the issue of proportionalization is still challenging on graphs. Finding a good concept representation or concept comparison method is very important for the quality of a matcher.

5.4 Summary

The contributions of this thesis are manifold: Detailed statistics about the WebIsA data set and its LOD derivatives, ALOD Classic and ALOD XL, were presented in terms of the size of the data sets and the distribution of relations within the networks. It could be shown that the distribution of degrees follows a power-law which is more pronounced in the ALOD XL data set. In addition, coverage

statistics have been calculated which indicate that the ALOD XL data set clearly outperforms DBpedia when it comes to concept coverages of nine ontologies.

Furthermore, two promising approaches have been presented that utilize hypernymy knowledge that was automatically extracted from the Web in order to calculate concept similarities: SPARQL-based hypernymy overlap and ALOD2Vec. ALOD2Vec is based on an adaption of the relatively new propositionalization approach RDF2Vec.

A simple method has been presented that allows to link labels to concepts on the Web and is additionally able to detect sub-concepts. The linker is not restricted to WebIsALOD but is also used for DBpedia and can be used for other data sets as well.

Using three gold standards, correlation results were presented which indicate that there is semantic knowledge contained in the knowledge graph and that the approaches presented perform competitively.

A synonymy gold standard was created comprising 1,234 instances of synonyms among monosemous concepts. Even though the gold standard turned out to be not helpful in the task of learning good concept similarity functions, it is publicly available and can be used for other research.

A full matcher was implemented and presented. The different concept similarity algorithms were individually evaluated within the ontology matching process. Using publicly available OAEI data sets as well as a corporate one, it could be shown that the approaches outperform string-based features and can add value within the matching process. However, current top-notch OAEI matchers could not be outperformed. To the knowledge of the author of this thesis, this is the very first approach utilizing the RDF2Vec method for the task of ontology matching. The matcher will also be evaluated in the upcoming OAEI campaign.

Moreover, an analysis was conducted in an enterprise environment, using real data models. In this context, a converter was presented that automatically generates ontologies from data models. These ontologies were matched using a gold standard provided by the company. An evaluation revealed that the matcher of this thesis performs close to one of the top-notch ontology matchers, namely LogMap. To the knowledge of the author of the present paper, this is the first time the LogMap matcher is used and evaluated in an enterprise environment. Even though, a fully automated matcher is not usable in a business scenario yet, a prototype was presented which allows to semantically query an existing data model and, thereby, speeding up the manual matching process. It could be demonstrated that the underlying similarity function of ontology matchers can help in assisting humans.

5.5 Outlook and Future Work

While matching with contextual information from the Web has a lot of potential, the current approaches are still in a very early stage.

Research on graph embeddings have made a big progress in recent years with new algorithm families, like word2vec-based approaches or translation-based approaches. Thus far, approaches have rarely been benchmarked against each other and no superior approach has been identified. It can be expected that the progress in this area will continue.

Similarly, not many ontology matchers use external knowledge other than thesauri. The area of how the Web can be used to derive meaningful semantic knowledge for the ontology matching process is still not well explored and interesting for further research.

The most promising approaches for concrete applications in the near-term will likely be semi-automatic approaches. Thus far, the human is still a crucial element in the process and will stay there for a foreseeable time. Even though most OAEI matchers are fully automated matchers, the performance in terms of F_1 -measure is still too low. Especially in an enterprise context, correctness is a *conditio sine qua non*, i.e., there must not be any mistakes in the alignments. Interactive approaches which allow for concrete user interaction with the matcher are already being explored [52]; the OAEI has an interactive track for evaluating such matchers since 2015.¹³ Within this area, another promising research direction is *active learning*. Here, the learner interacts with the human by asking her about concrete instances. This approach allows to train a good classifier by providing a very limited set of human annotations (compared to very large pre-built gold standards). [125, p. 1]

So far, current approaches do not cover the whole data integration process: Up to now, data translation rules still have to be defined manually without much assistance. Especially in an enterprise environment, this is a time-consuming process and an interesting research area for the future.

The focus of the present paper is on schema matching. Given that the ALOD data set – unlike thesauri – also contains named instances such as persons and places, the matcher might also be suited for instance matching or even combined schema and instance matching¹⁴.

Concerning the approach presented in this thesis, only one option for utilizing Web-scale knowledge has been pursued. There are three ways in which the current research focusing on this approach can be improved in the future: Firstly, more pro-

¹³see <http://sws.ifi.uio.no/oei/interactive/>

¹⁴In the OAEI 2018 campaign, a new track has been presented which combines schema and instance mapping and is based on the DBkWik data set [79]. See: <http://oei.ontologymatching.org/2018/knowledgegraph/index.html>

positionalization techniques for very large data sets could be explored. Secondly, the matcher itself can be enhanced to use more information available in ontologies. And lastly, the data sets to be used can be improved. WebIsALOD is currently the only Web-scale RDF data-set and still has some pitfalls such as the restriction to hypernymy relations and noise. More such data sets can be created in the future that address the issues described and, thereby, help to better utilize the knowledge contained in the Web.

Bibliography

- [1] Giovanni Acampora, Vincenzo Loia, and Autilia Vitiello. Enhancing Ontology Alignment Through a Memetic Aggregation of Similarity Measures. *Information Sciences*, 250:1–20, November 2013.
- [2] Dean Allemang and James A. Hendler. *Semantic Web for the Working Ontologist: Modeling in RDF, RDFS and OWL*. Morgan Kaufmann Publishers/Elsevier, Amsterdam; Boston, 2008.
- [3] Grant Allen and Michael Owens. *The Definitive Guide to SQLite*. The expert’s voice in open source. Apress, New York, NY, 2 edition, 2010.
- [4] Apache Software Foundation. Apache Jena - What is Jena? https://jena.apache.org/about_jena/about, 2018. Accessed: 2018-06-13.
- [5] Arvind Arasu and Josep Domingo-Ferrer. Record Matching. In M. Tamer Özsu and Ling Liu, editors, *Encyclopedia of Database Systems*, pages 2354–2358. Springer New York, New York, NY, 2009.
- [6] Günter Bamberg, Franz Baur, and Michael Krapp. *Statistik*. Lehr- und Handbücher der Wirtschafts- und Sozialwissenschaften. Oldenbourg, München, 17., überarb. Aufl. edition, 2012.
- [7] ING Bank. ING Bank Annual Report 2017, 2018.
- [8] Albert-Laszlo Barabasi and Reka Albert. Emergence of Scaling in Random Networks. *Science*, 286(5439):509–512, 1999.
- [9] Jakob Barion. *Philosophie: Einführung in ihre Terminologie und Hauptprobleme*. Bouvier, Bonn, 1977.
- [10] Basel Committee on Banking Supervision. *Principles for Effective Risk Data Aggregation and Risk Reporting*. Bank for Internat. Settlements, Basel, 2013.

- [11] Basel Committee on Banking Supervision. History of the Basel Committee. <https://www.bis.org/bcbs/history.htm>, 2018. Accessed: 2018-06-01.
- [12] Zohra Bellahsene, Angela Bonifati, and Erhard Rahm, editors. *Schema Matching and Mapping*. Data-Centric Systems and Applications. Springer, Heidelberg, 2011.
- [13] Massimo Benerecetti, Paolo Bouquet, and Chiara Ghidini. Contextual Reasoning Distilled. *Journal of Experimental & Theoretical Artificial Intelligence*, 12(3):279–305, July 2000.
- [14] Ron Bergers and Jasper Meijerink. The New Gold. <https://www2.deloitte.com/nl/nl/pages/data-analytics/articles/the-new-gold.html>, 2017. Accessed: 2018-08-12.
- [15] Tim Berners-Lee. Linked Data - Design Issues. <https://www.w3.org/DesignIssues/LinkedData.html>, 2006. Accessed: 2018-07-25.
- [16] Tim Berners-Lee. Semantic Web and Linked Data. [https://www.w3.org/2009/Talks/0120-campus-party-tbl/#\(1\)](https://www.w3.org/2009/Talks/0120-campus-party-tbl/#(1)), 2009. Accessed: 2018-06-20.
- [17] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. Translating Embeddings for Modeling Multi-Relational Data. In *NIPS'13 Proceedings of the 26th International Conference on Neural Information Processing Systems*, volume 2, pages 2787–2795, Lake Tahoe, Nevada, 2013.
- [18] Paolo Bouquet, Marc Ehrig, Jérôme Euzenat, Enrico Franconi, Pascal Hitzler, Markus Krotzsch, Luciano Serafini, Giorgos Stamou, York Sure, and Sergio Tessaris. Specification of a Common Framework for Characterizing Alignment. Deliverable D2.2.1, 2005.
- [19] Sandra Bracholdt, Volker Saggau, and Jan P. Portisch. View Building using Graph Theory and Black & White Decisions, July 2018. US Patent Application 16/027,010.
- [20] Dan Brickley and Libby Miller. FOAF Vocabulary Specification. <http://xmlns.com/foaf/spec/20140114.html>, 2014. Accessed: 2018-08-06.
- [21] Keith Brown and Jim Miller. *The Cambridge Dictionary of Linguistics*. Cambridge University Press, Cambridge; New York, 2013.

- [22] Elia Bruni, Gemma Boleda, Marco Baroni, and Nam-Khanh Tran. Distributional Semantics in Technicolor. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pages 136–145. Association for Computational Linguistics, 2012.
- [23] Elia Bruni, Nam-Khanh Tran, and Marco Baroni. Multimodal Distributional Semantics. *Journal of Artificial Intelligence Research (JAIR)*, 49(2014):1–47, 2014.
- [24] Fred Buckley and Marty Lewinter. *A Friendly Introduction to Graph Theory*. Prentice Hall, Upper Saddle River, N.J, 2003.
- [25] Alexander Budanitsky and Graeme Hirst. Evaluating Wordnet-based Measures of Lexical Semantic Relatedness. *Computational Linguistics*, 32(1):13–47, 2006.
- [26] Mario Bunge and Martin Mahner. *Über die Natur der Dinge: Materialismus und Wissenschaft*. S. Hirzel Verlag, Stuttgart, 2004.
- [27] Albert Busch and Oliver Stenschke, editors. *Germanistische Linguistik. Bachelor-Wissen*. Narr, Tübingen, 3., überarb. und erw. aufl edition, 2014.
- [28] Dietrich Busse. *Semantik*. Number 3280 in UTB Sprachwissenschaft. Fink, Paderborn, 2009.
- [29] Jeremy Carroll, Dave Reynolds, Ian Dickinson, Andy Seaborne, Chris Döllin, and Kevin Wilkinson. Jena: Implementing the Semantic Web Recommendations. In *Proceedings of the 13th International World Wide Web (WWW) Conference*, pages 74–83, New York, NY, 2004.
- [30] Peter Christen. *Data Matching: Concepts and Techniques for Record Linkage, Entity Resolution, and Duplicate Detection*. Data-Centric Systems and Applications. Springer, Berlin, 2012.
- [31] Rudi Cilibrasi and Paul M. B. Vitányi. The Google Similarity Distance. *IEEE Transactions on Knowledge and Data Engineering*, 19(3):370–383, 2007.
- [32] Michael Cochez, Petar Ristoski, Simone Paolo Ponzetto, and Heiko Paulheim. Biased Graph Walks for RDF Graph Embeddings. In *Proceedings of the 7th International Conference on Web Intelligence, Mining and Semantics*, Amantea, Italy, 2017. ACM Press.

- [33] Ronan Collobert and Jason Weston. A Unified Architecture for Natural Language Processing: Deep Neural Networks with Multitask Learning. In *Learning*, *Proceedings of the 25 th International Conference on Machine Learning*, pages 160–167, 2008.
- [34] David Crystal. *The Cambridge Encyclopedia of the English Language*. Cambridge University Press, Cambridge, U.K.; New York, 2 edition, 2003.
- [35] Jérôme David, Jérôme Euzenat, François Scharffe, and Cássia Trojahn dos Santos. The Alignment API 4.0. *Semantic Web Journal*, 2(1):3–10, 2011.
- [36] Jos de Bruijn, Marc Ehrig, Christina Feier, Francisco Martín-Recuerda, François Scharffe, and Moritz Weiten. Ontology Mediation, Merging, and Aligning. In John Davies, Rudi Studer, and Paul Warren, editors, *Semantic Web Technologies: Trends and Research in Ontology-Based Systems*, pages 95–113. Wiley and Sons, Chichester, England, 2006.
- [37] Jos de Bruijn, Douglas Foxvog, and Kerstin Zimmerman. D4.3.1 Ontology Mediation Patterns Library V1, 2005.
- [38] Gerben K. D. de Vries. A Fast Approximation of the Weisfeiler-Lehman Graph Kernel for RDF Data. In David Hutchison, Takeo Kanade, Josef Kittler, Jon M. Kleinberg, Friedemann Mattern, John C. Mitchell, Moni Naor, Oscar Nierstrasz, C. Pandu Rangan, Bernhard Steffen, Madhu Sudan, Demetri Terzopoulos, Doug Tygar, Moshe Y. Vardi, Gerhard Weikum, Hendrik Blockeel, Kristian Kersting, Siegfried Nijssen, and Filip Železný, editors, *Machine Learning and Knowledge Discovery in Databases*, volume 8188, pages 606–621. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013.
- [39] Gerben Klaas Dirk de Vries and Steven de Rooij. Substructure Counting Graph Kernels for Machine Learning From RDF Data. *Web Semantics: Science, Services and Agents on the World Wide Web*, 35:71–84, December 2015.
- [40] Lee R. Dice. Measures of the Amount of Ecologic Association Between Species. *Ecology*, 26(3):297–302, July 1945.
- [41] Warith Eddine Djeddi and Mohamed Tarek Khadir. A Novel Approach Using Context-Based Measure for Matching Large Scale Ontologies. In Ladjel Bellatreche and Mukesh Mohania, editors, *Data Warehousing and Knowledge Discovery: 16th International Conference, DAWAK 2014, Munich, Germany, September 2-4, 2014. Proceedings*, Lecture Notes in Computer Science, pages 320–331, New York, 2014. Springer.

- [42] Kien Do, Truyen Tran, and Svetha Venkatesh. Knowledge Graph Embedding with Multiple Relation Projections. *arXiv:1801.08641 [cs]*, January 2018.
- [43] AnHai Doan. Best-Effort Data Integration. In *NSF Workshop on Data Integration*, 2006.
- [44] AnHai Doan, Alon Halevy, and Zachary G. Ives. *Principles of Data Integration*. Morgan Kaufmann, Waltham, MA, 2012.
- [45] Zlatan Dragisic, Kai Eckert, Jérôme Euzenat, Daniel Faria, Alfio Ferrara, Roger Granada, Valentina Ivanova, Ernesto Jiménez-Ruiz, Andreas Oskar Kempf, Patrick Lambrix, Stefano Montanelli, Heiko Paulheim, Dominique Ritze, Pavel Shvaiko, Alessandro Solimando, and Cássia Trojahn. Results of the Ontology Alignment Evaluation Initiative 2014, 2014.
- [46] M. Duerst and M. Suignard. Internationalized Resource Identifiers (IRIs) - RFC 3987. *The Internet Society*, 2005.
- [47] Marc Ehrig. *Ontology Alignment: Bridging the Semantic Gap*. Number 4 in Semantic Web and Beyond. Springer, New York, 2007.
- [48] Elearn. *Information and Knowledge Management*. Elsevier/Pergamon Flexible Learning, Amsterdam; Boston, 2009.
- [49] Jérôme Euzenat and Pavel Shvaiko. *Ontology Matching*. Springer, New York, 2nd edition, 2013.
- [50] Jérôme Euzenat, François Scharffe, and Antoine Zimmermann. Expressive Alignment Language and Implementation, 2007.
- [51] Muhammad Fahad. ER2OWL: Generating OWL Ontology from ER Diagram. In Zhongzhi Shi, E. Mercier-Laurent, and D. Leake, editors, *Intelligent Information Processing IV*, volume 288, pages 28–37. Springer US, Boston, MA, 2008.
- [52] Sean M. Falconer and Natalya F. Noy. Interactive Techniques to Support Ontology Matching. In Zohra Bellahsene, Angela Bonifati, and Erhard Rahm, editors, *Schema Matching and Mapping*, Data-Centric Systems and Applications. Springer, Heidelberg, 2011.
- [53] Lev Finkelstein, Evgeniy Gabrilovich, Yossi Matias, Ehud Rivlin, Zach Solan, Gadi Wolfman, and Eytan Ruppín. Placing Search in Context: The Concept Revisited. In *Proceedings of the 10th international conference on World Wide Web*, pages 406–414. ACM, 2001.

- [54] Valeria Fonti and Eduard Belitser. Feature Selection using LASSO, 2017.
- [55] Forbes. The World’s Biggest Public Companies. <https://www.forbes.com/global2000/list/>, 2017. Accessed: 2018-08-12.
- [56] Aldo Gangemi, Nicola Guarino, Claudio Masolo, Alessandro Oltramari, and Luc Schneider. Sweetening Ontologies with DOLCE. In *International Conference on Knowledge Engineering and Knowledge Management*, pages 166–181. Springer, Berlin, Heidelberg, 2002.
- [57] Chiara Ghidini, Luciano Serafini, and Sergio Tessaris. On Relating Heterogeneous Elements from Different Ontologies. In Boicho Kokinov, Daniel C. Richardson, Thomas R. Roth-Berghofer, and Laure Vieu, editors, *Modeling and Using Context*, volume 4635, pages 234–247. Springer Berlin Heidelberg, Berlin, Heidelberg, 2007.
- [58] Bernardo Cuenca Grau, Zlatan Dragisic, Kai Eckert, Jérôme Euzenat, Alfio Ferrara, Roger Granada, Valentina Ivanova, Ernesto Jiménez-Ruiz, Andreas Oskar Kempf, Patrick Lambrix, Andriy Nikolov, Heiko Paulheim, Dominique Ritze, François Scharffe, Pavel Shvaiko, Cássia Trojahn, and Ondrej Zamazal. Results of the Ontology Alignment Evaluation Initiative 2013. Sydney, Australia, 2013.
- [59] Aditya Grover and Jure Leskovec. node2vec: Scalable Feature Learning for Networks. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 855–864. ACM Press, 2016.
- [60] Thomas Gärtner, Tamás Horváth, and Stefan Wrobel. Graph Kernels. In Claude Sammut and Geoffrey I. Webb, editors, *Encyclopedia of Machine Learning*. Springer, Boston, MA, 2011.
- [61] Thomas R. Gruber. Toward Principles for the Design of Ontologies Used for Knowledge Sharing. *International Journal of Human-Computer Studies*, 43(5-6):907–928, 1993.
- [62] Adil Hameed, Alun Preece, and Derek Sleeman. Ontology Reconciliation. In Steffen Staab and Rudi Studer, editors, *Handbook on Ontologies*, pages 231–250. Springer Berlin Heidelberg, Berlin, Heidelberg, 2004.
- [63] Birgit Hamp and Helmut Feldweg. GermaNet - a Lexical-Semantic Net for German. In *Proceedings of the ACL workshop Automatic Information Extraction and Building of Lexical Semantic Resources for NLP Applications*, Madrid, Spain, 1997.

- [64] Stevan Harnad. The Symbol Grounding Problem. *CoRR*, cs.AI/9906002, 1999.
- [65] Zellig S. Harris. Distributional Structure. *WORD*, 10(2-3):146–162, August 1954.
- [66] Marti A. Hearst. Automatic Acquisition of Hyponyms from Large Text Corpora. volume 2, pages 539–545. Association for Computational Linguistics, 1992.
- [67] Jim Hendler. Web 3.0 Emerging. *Computer*, 42(1):111–113, 2009.
- [68] Nicolaus Henke, Jacques Bughin, Michael Chui, James Manyika, Tamim Saleh, Bill Wiseman, and Guru Sethupathy. The age of Analytics: Competing in a Data-Driven World, 2016.
- [69] Verena Henrich and Erhard Hinrichs. GernEdiT – The GermaNet Editing Tool. In *Proceedings of the Seventh Conference on International Language Resources and Evaluation (LREC 2010)*, pages 2228–2235, Valletta, Malta, 2010.
- [70] Martin Hepp, Pieter De Leenheer, Aldo de Moor, and York Sure, editors. *Ontology Management: Semantic Web, Semantic Web Services, and Business Applications*. Number 7 in Semantic Web and Beyond. Springer, New York, NY, 2008.
- [71] Sven Hertling. Hertuda Results for OEAI 2012. In *Proceedings of the 7th International Conference on Ontology Matching*, volume 4, pages 141–144, Graz, Austria, 2012.
- [72] Sven Hertling. WikiV3 Results for OAEI 2017. In Pavel Shvaiko, Jérôme Euzenat, Ernesto Jiménez-Ruiz, Michelle Cheatham, and Oktie Hassanzadeh, editors, *OM-2017: Proceedings of the Twelfth International Workshop on Ontology Matching*, pages 190–195, Wien, Austria, 2017.
- [73] Sven Hertling and Heiko Paulheim. WikiMatch - Using Wikipedia for Ontology Matching. In Pavel Shvaiko, Jérôme Euzenat, Anastasios Kementsietsidis, Ming Mao, Natasha Noy, and Heiner Stuckenschmidt, editors, *OM-2012: Proceedings of the ISWC Workshop*, pages 37–48, 2012.
- [74] Sven Hertling and Heiko Paulheim. WikiMatch – Using Wikipedia for Ontology Matching. 2012.

- [75] Sven Hertling and Heiko Paulheim. WikiMatch Results for OEAI 2012. In Pavel Shvaiko, Jérôme Euzenat, Anastasios Kementsietsidis, Ming Mao, Natasha Noy, and Heiner Stuckenschmidt, editors, *OM-2012: Proceedings of the ISWC Workshop*, pages 213–219, 2012.
- [76] Sven Hertling and Heiko Paulheim. WebIsALOD: Providing Hypernymy Relations Extracted From the Web as Linked Open Data. In *International Semantic Web Conference*, pages 111–119. Springer, 2017.
- [77] Felix Hill, Roi Reichart, and Anna Korhonen. SimLex-999: Evaluating Semantic Models with (Genuine) Similarity Estimation. *arXiv:1408.3456 [cs]*, 41(4):665–695, August 2014.
- [78] Pascal Hitzler, Markus Krötzsch, Sebastian Rudolph, and York Sure. *Semantic Web: Grundlagen*. eXamen.press. Springer, Berlin, 1 edition, 2008.
- [79] Alexandra Hofmann, Samresh Perchani, Jan Portisch, Sven Hertling, and Heiko Paulheim. DBkWik: Towards Knowledge Graph Creation from Thousands of Wikis. 2017.
- [80] David Holmes and M.Catherine McCabe. Improving Precision and Recall for Soundex Retrieval. pages 22–26. IEEE Comput. Soc, 2002.
- [81] Ian Horrocks, Peter F. Patel-Schneider, Harold Boley, Said Tabet, Benjamin Grosf, and Mike Dean. SWRL A Semantic Web Rule Language Combining OWL and RuleML, 2004.
- [82] Eric H. Huang, Richard Socher, Christopher D. Manning, and Andrew Y. Ng. Improving Word Representations via Global Context and Multiple Word Prototypes. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pages 873–882. Association for Computational Linguistics, 2012.
- [83] Internet Live Stats. Total Number of Websites. <http://www.internetlivestats.com/total-number-of-websites/>, 2018. Accessed: 2018-06-13.
- [84] Prateek Jain, Pascal Hitzler, Amit P. Sheth, Kunal Verma, and Peter Z. Yeh. Ontology Alignment for Linked Open Data. In Peter F. Patel-Schneider, Yue Pan, Pascal Hitzler, Peter Mika, Lei Zhang, Jeff Z. Pan, Ian Horrocks, and Birte Glimm, editors, *The Semantic Web – ISWC 2010*, volume 6496, pages 402–417. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010.

- [85] Guoliang Ji, Shizhu He, Liheng Xu, Kang Liu, and Jun Zhao. Knowledge Graph Embedding via Dynamic Mapping Matrix. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, volume 1, pages 687–696, 2015.
- [86] Yantao Jia, Yuanzhuo Wang, Hailun Lin, Xiaolong Jin, and Xueqi Cheng. Locally Adaptive Translation for Knowledge Graph Embedding. In *Proceedings of the 30th AAAI Conference on Artificial Intelligence (AAAI'16)*, pages 992–998, 2016.
- [87] Ernesto Jiménez-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 – ISWC 2011*, volume 7031, pages 273–288. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011.
- [88] Ernesto Jiménez-Ruiz, Bernardo Cuenca Grau, and Valerie Cross. LogMap Family Results for OAEI 2015. In Pavel Shvaiko, Jérôme Euzenat, Ernesto Jiménez-Ruiz, Michelle Cheatham, and Oktie Hassanzadeh, editors, *Ontology Matching OM-2015: Proceedings fo the ISWC Workshop*, pages 171–175, Bethlehem, PA USA, 2015.
- [89] Ernesto Jiménez-Ruiz, Bernardo Cuenca Grau, and Valerie Cross. LogMap Family Participation in the OAEI 2016. In Pavel Shvaiko, Jérôme Euzenat, Ernesto Jiménez-Ruiz, Michelle Cheatham, Oktie Hassanzadeh, and Ryutaro Ichise, editors, *Ontology Matching OM-2016: Proceedings of the ISWC Workshop*, pages 185–189, Kōbe, Japan, 2016.
- [90] Ernesto Jiménez-Ruiz, Bernardo Cuenca Grau, and Valerie Cross. LogMap Family Participation in the OAEI 2017. In Pavel Shvaiko, Jérôme Euzenat, Ernesto Jiménez-Ruiz, Michelle Cheatham, and Oktie Hassanzadeh, editors, *OM-2017: Proceedings of the Twelfth International Workshop on Ontology Matching*, pages 153–157, Wien, Austria, 2017.
- [91] Yuzhe Jin, Emre Kıcıman, Kuansan Wang, and Ricky Loynd. Entity Linking at the Tail: Sparse Signals, Unknown Entities, and Phrase Models. pages 453–462. ACM Press, 2014.
- [92] Marouen Kachroudi, Gayo Diallo, and Sadok Ben Yahia. OAEI 2017 results of KEPLER. In Pavel Shvaiko, Jérôme Euzenat, Ernesto Jiménez-Ruiz,

- Michelle Cheatham, and Oktie Hassanzadeh, editors, *OM-2017: Proceedings of the Twelfth International Workshop on Ontology Matching*, pages 138–145, Wien, Austria, 2017.
- [93] Anne Kao and Stephen R. Poteet, editors. *Natural Language Processing and Text Mining*. Springer, London, 2007.
- [94] Anne Kao and Stephen R. Poteet. Overview. In Anne Kao and Stephen R. Poteet, editors, *Natural Language Processing and Text Mining*, pages 1–8. Springer, London, 2007.
- [95] Mansoor Ahmed Khan, Gunnar Aastrand Grimnes, and Andreas Dengel. Two Pre-Processing Operators for Improved Learning From Semantic Web Data, 2010.
- [96] Douwe Kiela and Stephen Clark. A Systematic Study of Semantic Vector Space Model Parameters. In *Proceedings of the 2nd Workshop on Continuous Vector Space Models and their Compositionality (CVSC)*, pages 21–30, 2014.
- [97] Won Kim and Jungyun Seo. Classifying Schematic and Data Heterogeneity in Multidatabase Systems. *Computer*, 24(12):12–18, December 1991.
- [98] Kai-Oliver Klauck and Claus Stegmann, editors. *Basel III: Vom regulatorischen Rahmen zu einer risikoadäquaten Gesamtbanksteuerung*. Schäffer-Poeschel Verlag, Stuttgart, 2012.
- [99] Michael Klein. Combining and Relating Ontologies: An Analysis of Problems and Solutions. In Asunción Gómez-Pérez, Michael Gruninger, Heiner Stuckenschmidt, and Michael Uschold, editors, *Proceedings of the IJCAI-01 Workshop on Ontologies and Information Sharing*, pages 53–62, Seattle, USA, 2001.
- [100] Jan Kotek. MapDB Release 2.0. <http://www.mapdb.org/download/mapdb-manual-20.pdf>, 2016. Accessed: 2018-06-01.
- [101] Stefan Kramer, Nada Lavrač, and Peter Flach. Propositionalization Approaches To Relational Data Mining. In Sašo Džeroski and Nada Lavrač, editors, *Relational Data Mining*, pages 262–291. Springer, Berlin; New York, 2001.
- [102] Jay A. Kreibich. What Is SQLite? In *Using SQLite*. O’Reilly Media, Inc., August 2010.

- [103] Thomas K Landauer and Susan T Dutnais. A Solution to Plato's Problem: The Latent Semantic Analysis Theory of Acquisition, Induction, and Representation of Knowledge. *Psychological Review*, 104(2):211–240, 1997.
- [104] Sebastian Löbner. *Semantik: Eine Einführung*. De Gruyter Studienbuch. de Gruyter Mouton, Berlin, 2nd edition edition, 2015.
- [105] Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N Mendes, Sebastian Hellmann, Mohamed Morsey, and Christian Bizer. DBpedia – A Large-scale, Multilingual Knowledge Base Extracted from Wikipedia. *Semantic Web*, 6(2):167–195, 2012.
- [106] Oliver Lehmborg, Robert Meusel, and Christian Bizer. Graph Structure in the Web: Aggregated by Pay-Level Domain. pages 119–128, Bloomington, USA, 2014. ACM Press.
- [107] Franz Lehner. *Wissensmanagement: Grundlagen, Methoden und technische Unterstützung*. Hanser, München, 5th edition, 2014.
- [108] Douglas B Lenat. *Building large knowledge-based systems : representation and inference in the Cyc Project*. Addison-Wesley, Reading, Mass., Bonn, 1. print. edition, 1990.
- [109] Vladimir Levenshtein. Binary Codes Capable of Correcting Deletions, Insertions, and Reversals. *Soviet Physics Doklady*, 10(8):707–710, 1966.
- [110] Feiyu Lin, Jonathan Butters, Kurt Sandkuhl, and Fabio Ciravegna. Context-based Ontology Matching: Concept and Application Cases. pages 1292–1298. IEEE, June 2010.
- [111] Feiyu Lin and Andrew Krizhanovsky. Multilingual Ontology Matching based on Wiktionary Data Accessible via SPARQL Endpoint. In *Proceedings of the 13th Russian Conf. on Digital Libraries RCDL'2011*, pages 19–26, Voronezh, Russia, 2011.
- [112] Bing Liu. *Web Data Mining: Exploring Hyperlinks, Contents, and Usage Data*. Data-centric systems and applications. Springer, Heidelberg; New York, 2 edition, 2011.
- [113] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval*, volume 1. Cambridge University Press Cambridge, Cambridge, England, 2008.

- [114] Viviana Mascardi, Angela Locoro, and Paolo Rosso. Automatic Ontology Matching Via Upper Ontologies: A Systematic Evaluation. *IEEE Transactions on Knowledge and Data Engineering*, pages 609–623, 2010.
- [115] Ken McRae, George S. Cree, Mark S. Seidenberg, and Chris McNorgan. Semantic Feature Production Norms for a Large Set of Living and Nonliving Things. *Behavior Research Methods*, 37(4):547–559, 2005.
- [116] Rainer Merkt. Semantische, methodische und prozessuale Integration - Theoretisch-konzeptionelle Ansätze und ihre Anwendung in Banken. In Kai-Oliver Klauck and Claus Stegmann, editors, *Basel III: Vom regulatorischen Rahmen zu einer risikoadäquaten Gesamtbanksteuerung*, pages 311–345. Schäffer-Poeschel Verlag, Stuttgart, 2012.
- [117] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient Estimation of Word Representations in Vector Space. *arXiv preprint arXiv:1301.3781*, 2013.
- [118] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S. Corrado, and Jeff Dean. Distributed Representations of Words and Phrases and Their Compositionality. In *Advances in Neural Information Processing Systems*, pages 3111–3119, 2013.
- [119] George A. Miller. WordNet: A Lexical Database for English. *Communications of the ACM*, 38(11):39–41, 1995.
- [120] Kevin P. Murphy. *Machine Learning: A Probabilistic Perspective*. Adaptive Computation and Machine Learning Series. MIT Press, Cambridge, MA, 2012.
- [121] Venkata Narasimha, Pavan Kappara, Ryutaro Ichise, and O P Vyas. LiDDM: A Data Mining System for Linked Data. *Linked Data On the Web*, 2011.
- [122] Daniel Nations. Is Web 3.0 Really a Thing? <https://www.lifewire.com/what-is-web-3-0-3486623>, 2018. Accessed: 2018-06-20.
- [123] Ian Niles and Adam Pease. Towards a Standard Upper Ontology. pages 2–9. ACM Press, 2001.
- [124] Daniel Oberle. How Ontologies Benefit Enterprise Applications, 2009.
- [125] Fredrik Olsson. A Literature Survey of Active Machine Learning in the Context of Natural Language Processing, 2009.

- [126] Oxford Dictionaries. How many words are there in the English language? <https://en.oxforddictionaries.com/explore/how-many-words-are-there-in-the-english-language/>, 2018. Accessed: 2018-06-13.
- [127] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The PageRank Citation Ranking: Bringing Order to the Web, 1998.
- [128] Heiko Paulheim. WeSeE-Match Results for OAEI 2012. In Pavel Shvaiko, Jérôme Euzenat, Anastasios Kementsietsidis, Ming Mao, Natasha Noy, and Heiner Stuckenschmidt, editors, *OM-2012: Proceedings of the ISWC Workshop*, pages 213–219, Boston, MA, 2012.
- [129] Heiko Paulheim and Sven Hertling. WeSeE-Match Results for OAEI 2013. In Pavel Shvaiko, Jérôme Euzenat, Kavitha Srinivas, Ming Mao, and Ernesto Jiménez-Ruiz, editors, *Ontology Matching OM-2013: Proceedings of the ISWC Workshop*, Sydney, Australia, 2013.
- [130] Ingo Plag, Sabine Lappe, Maria Braun, and Mareile Schramm. *Introduction to English Linguistics*. De Gruyter Mouton textbook. De Gruyter Mouton, Berlin, third, revised and enlarged edition edition, 2015.
- [131] Martin F. Porter. An Algorithm for Suffix Stripping. *Program*, 14(3):130–137, 1980.
- [132] Hans J. Postel. Die Kölner Phonetik. Ein Verfahren zur Identifizierung von Personennamen auf der Grundlage der Gestaltanalyse. *IBM-Nachrichten*, 19:925–931, 1969.
- [133] Princeton University. Princeton University | WordNet | doorhandle. <http://wordnetweb.princeton.edu/perl/webwn?o2=&o0=1&o8=1&o1=1&o7=&o5=&o9=&o6=&o3=&o4=&s=doorhandle>, 2010. Accessed: 2018-06-16.
- [134] Princeton University. Princeton University "About WordNet". <https://wordnet.princeton.edu/>, 2010. Accessed: 2018-06-16.
- [135] Gilbert J. B. Probst, Steffen P. Raub, and Kai Romhardt. *Wissen managen: Wie Unternehmen ihre wertvollste Ressource optimal nutzen*. Springer Gabler, Wiesbaden, 7. auflage edition, 2012.
- [136] Erhard Rahm and Philip A. Bernstein. A Survey of Approaches to Automatic Schema Matching. *The VLDB Journal*, 10(4):334–350, December 2001.

- [137] Edie Rasmussen. Stoplists. In Ling Liu and M. Tamer Özsu, editors, *Encyclopedia of Database Systems*, pages 2794–2796. Springer New York, New York, NY, 2009.
- [138] Michael Röder, Axel-Cyrille Ngonga Ngomo, and Martin Strohbach. Deliverable 2.1: Detailed Architecture of the HOBBIT Platform, 2016.
- [139] Radim Řehůřek and Petr Sojka. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50, Valletta, Malta, May 2010. ELRA.
- [140] Reuters. Deutsche-Bank-IT-Chefin nach umstrittenen Äußerungen unter Druck. *Reuters*, March 2018. Accessed: 2018-07-08.
- [141] Nick Riemer. *Introducing Semantics*. Cambridge Introductions to Language and Linguistics. Cambridge University Press, Cambridge; New York, 2015.
- [142] Marek Ristock. Transformation in die IT einer Bank. In Kai-Oliver Klauck and Claus Stegmann, editors, *Basel III: Vom regulatorischen Rahmen zu einer risikoadäquaten Gesamtbanksteuerung*, pages 299–310. Schäffer-Poeschel Verlag, Stuttgart, 2012.
- [143] Petar Ristoski. *Exploiting Semantic Web Knowledge Graphs in Data Mining*. PhD thesis, Universität Mannheim, Mannheim, 2017.
- [144] Petar Ristoski, Christian Bizer, and Heiko Paulheim. Mining the Web of Linked Data with RapidMiner. *Web Semantics: Science, Services and Agents on the World Wide Web*, 35:142–151, December 2015.
- [145] Petar Ristoski and Heiko Paulheim. A Comparison of Propositionalization Strategies for Creating Features from Linked Open Data. In *Proceedings of the 1st Workshop on Linked Data for Knowledge Discovery*, volume 1232, pages 6–15, Nancy, 2014.
- [146] Petar Ristoski and Heiko Paulheim. RDF2vec: RDF Graph Embeddings for Data Mining. In *International Semantic Web Conference*, pages 498–514. Springer, 2016.
- [147] Petar Ristoski, Jessica Rosati, Tommaso Di Noia, Renato De Leone, and Heiko Paulheim. RDF2vec: RDF Graph Embeddings and Their Applications. *Semantic Web Journal*, 2017.

- [148] Claude Sammut and Geoffrey I. Webb, editors. *Encyclopedia of Machine Learning and Data Mining*. Springer, Boston, MA, 2017.
- [149] SAP SE. Feature Scope Description for SAP Financial Services Data Management, 2017.
- [150] SAP SE. Customizing and Extending PowerDesigner, 2018.
- [151] SAP SE. SAP Annual Report 2017 on Form 20-F, 2018.
- [152] Ferdinand de Saussure, Charles Bally, Albert Riedlinger, Herman Lommel, and Peter Ernst. *Grundfragen der allgemeinen Sprachwissenschaft*. DeGruyter-Studienbuch. de Gruyter, Berlin, 3rd edition, 2001.
- [153] Julian Seitner, Christian Bizer, Kai Eckert, Stefano Faralli, Robert Meusel, Heiko Paulheim, and Simone Paolo Ponzetto. A Large DataBase of Hypernymy Relations Extracted from the Web. In *LREC*, 2016.
- [154] Nino Shervashidze, Pascal Schweitzer, Erik Jan van Leeuwen, Kurt Mehlhorn, and Karsten M. Borgwardt. Weisfeiler-Lehman Graph Kernels. *Journal of Machine Learning Research*, 12:2539–2561, 2011.
- [155] Amit Sheth and Vipul Kashyap. So Far (Schematically) yet So Near (Semantically). In *Interoperable Database Systems (Ds-5)*, pages 283–312. Elsevier, 1993.
- [156] Pavel Shvaiko and Jérôme Euzenat. Ten Challenges for Ontology Matching. In Robert Meersman and Zahir Tari, editors, *On the Move to Meaningful Internet Systems: OTM 2008*, volume 5332, pages 1164–1182. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008.
- [157] Pavel Shvaiko and Jérôme Euzenat. A Survey of Schema-Based Matching Approaches. In David Hutchison, Takeo Kanade, Josef Kittler, Jon M. Kleinberg, Friedemann Mattern, John C. Mitchell, Moni Naor, Oscar Nierstrasz, C. Pandu Rangan, Bernhard Steffen, Madhu Sudan, Demetri Terzopoulos, Dough Tygar, Moshe Y. Vardi, Gerhard Weikum, and Stefano Spaccapietra, editors, *Journal on Data Semantics IV*, volume 3730, pages 146–171. Springer Berlin Heidelberg, Berlin, Heidelberg, 2005.
- [158] Pavel Shvaiko and Jérôme Euzenat. Ontology Matching: State of the Art and Future Challenges. *IEEE Transactions on Knowledge and Data Engineering*, 25(1):158–176, January 2013.

- [159] Carina Silberer, Vittorio Ferrari, and Mirella Lapata. Visually Grounded Meaning Representations. *IEEE Trans. Pattern Anal. Mach. Intell.*, 39(11):2284–2297, 2017.
- [160] Arvind Singh. Is Big Data the New Black Gold? <https://www.wired.com/insights/2013/02/is-big-data-the-new-black-gold/>, 2013. Accessed: 2018-08-11.
- [161] Alessandro Solimando, Ernesto Jiménez-Ruiz, and Giovanna Guerrini. Detecting and Correcting Conservativity Principle Violations in Ontology-to-Ontology Mappings. In Peter Mika, Tania Tudorache, Abraham Bernstein, Chris Welty, Craig Knoblock, Denny Vrandečić, Paul Groth, Natasha Noy, Krzysztof Janowicz, and Carole Goble, editors, *The Semantic Web – ISWC 2014*, volume 8797, pages 1–16. Springer International Publishing, Cham, 2014.
- [162] Alessandro Solimando, Ernesto Jiménez-Ruiz, and Giovanna Guerrini. Minimizing Conservativity Violations in Ontology Alignments: Algorithms and Evaluation. *Knowledge and Information Systems*, 51(3):775–819, June 2017.
- [163] Peter Spyns and Mustafa Jarrar. Data Modelling versus Ontology Engineering, 2002.
- [164] Steffen Staab and Rudi Studer, editors. *Handbook on Ontologies*. International Handbooks on Information Systems. Springer Berlin Heidelberg, Berlin, Heidelberg, 2 edition, 2009.
- [165] Standard Upper Ontology Working Group. Standard Upper Ontology Working Group (SUO WG) - Home Page. <http://web.archive.org/web/20140512225349/http://suo.ieee.org/>, 2003. Accessed: 2018-06-13.
- [166] Giorgos Stoilos, Giorgos Stamou, and Stefanos Kollias. A String Metric for Ontology Alignment. In David Hutchison, Takeo Kanade, Josef Kittler, Jon M. Kleinberg, Friedemann Mattern, John C. Mitchell, Moni Naor, Oscar Nierstrasz, C. Pandu Rangan, Bernhard Steffen, Madhu Sudan, Demetri Terzopoulos, Dough Tygar, Moshe Y. Vardi, Gerhard Weikum, Yolanda Gil, Enrico Motta, V. Richard Benjamins, and Mark A. Musen, editors, *The Semantic Web – ISWC 2005*, volume 3729, pages 624–637. Springer Berlin Heidelberg, Berlin, Heidelberg, 2005.

- [167] Tobias Swoboda, Matthias Hemmje, Mihai Dascalu, and Stefan Trausan-Matu. Combining Taxonomies using Word2vec. pages 131–134. ACM Press, 2016.
- [168] The Economist. Fuel of the Future - Data is Giving Rise to a New Economy. *The Economist*, May 2017.
- [169] The Economist. The World’s Most Valuable Resource. *The Economist*, 2017.
- [170] W3C. Extensible Markup Language (XML) 1.1 (Second Edition). <https://www.w3.org/TR/xml11/>, 2006. Accessed: 2018-06-20.
- [171] W3C. OWL 2 Web Ontology Language Structural Specification and Functional-Style Syntax (Second Edition). <https://www.w3.org/TR/2012/REC-owl2-syntax-20121211/>, 2012. Accessed: 2018-06-21.
- [172] W3C. SPARQL 1.1 Protocol. <https://www.w3.org/TR/2013/REC-sparql11-protocol-20130321/>, 2013.
- [173] W3C. JSON-LD 1.0. <https://www.w3.org/TR/json-ld/>, 2014. Accessed: 2018-06-28.
- [174] W3C. RDF 1.1 N-Quads. <https://www.w3.org/TR/n-quads/>, 2014. Accessed: 2018-06-28.
- [175] W3C. RDF 1.1 Primer. <https://www.w3.org/TR/rdf11-primer/>, 2014. Accessed: 2018-06-14.
- [176] W3C. RDF 1.1 Turtle. <https://www.w3.org/TR/turtle/>, 2014. Accessed: 2018-06-28.
- [177] Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. Knowledge Graph Embedding by Translating on Hyperplanes. pages 1112–1119, 2014.
- [178] Matteo Gianpietro Zago. Why the Web 3.0 Matters and you should know about it. <https://medium.com/@matteozago/why-the-web-3-0-matters-and-you-should-know-about-it-a5851d63c949>, 2018. Accessed: 2018-06-20.
- [179] Ondřej Zamazal and Vojtěch Svátek. The Ten-Year OntoFarm and its Fertilization within the Onto-Sphere. *Web Semantics: Science, Services and Agents on the World Wide Web*, 43:46–53, March 2017.

- [180] Yuanzhe Zhang, Xuepeng Wang, Siwei Lai, Shizhu He, Kang Liu, Jun Zhao, and Xueqiang Lv. Ontology Matching with Word Embeddings. In Maosong Sun, Yang Liu, and Jun Zhao, editors, *Chinese Computational Linguistics and Natural Language Processing Based on Naturally Annotated Big Data*, volume 8801, pages 34–45. Springer International Publishing, Cham, 2014.

Appendix A

Program Code / Resources

In the following, smaller code examples are listed. Note that the full implementation is not listed here due to its size. For all the coding refer to the CD attached to this thesis.

A.1 ALOD Modification Sequence

The following describes the sequence of string modifications applied to a label in order to link it to the ALOD data set. The modifications are executed in ascending order and get more aggressive.

1. `LowerCaseModifier`
Lowercases the label; *European Union*, for instance, is transformed to *european union*.
2. `TokenizeSpaceSeparateLowercaseModifier`
This modifier tokenizes a label and concatenates the tokens using spaces. Leading and trailing spaces are removed and the label is lowercased. An example would be *EuropeanUnion* which is transformed to *european union*.
3. `CharactersOnlyTokenizeSpaceSeparateLowercaseModifier`
This modifier will delete all non-ASCII characters and then process the label further like the `TokenizeSpaceSeparateLowercaseModifier`. An example would be *TreatyOfLisbon_2007* which would be translated into *treaty of lisbon*.

A.2 Python Code: Word2vec Most Related Concepts

The code presented in listing A.1 is used to retrieve the most similar concepts for a given user input.

```
1 from gensim.models import KeyedVectors
2
3 # the path to the pre-trained word2vec vectors
4 path_to_vector_file = 'C://GoogleNews-vectors-
   negative300.bin'
5
6 # loading the vectors
7 word_vectors = KeyedVectors.load_word2vec_format\
8     (path_to_vector_file, binary=True)
9
10 # allowing the user to search for the topn similar
11 # words using the console
12 # for exiting the user can write 'exit' and hit
   enter
13 user_input = ''
14 while(user_input != 'exit'):
15     user_input = input('Next Word:')
16     print('Your Input: ' + user_input)
17
18     try:
19         result = word_vectors.most_similar(
20             user_input, topn=15)
21         for word, score in result:
22             print(word + " (" + str(score) + ")")
23         print('\r\n')
24     except KeyError:
25         print('Term not found.\r\n')
```

Listing A.1: Code to Retrieve Closest Concepts

Appendix B

Further Experimental Results

In the following, further experimental results are listed.

B.1 Coverage Statistics: DBpedia vs. ALOD Classic

Dataset		DBpedia		WebIsALOD	
Name	# of Terms (English)	Whole Term	Tokens	Whole Term	Tokens
Anatomy	11,928	0.17010	<u>0.85680</u>	<u>0.20154</u>	0.62332
Conference	488	0.18647	0.85450	<u>0.26639</u>	<u>0.87909</u>
BioMed	408,483	<u>0.10749</u>	<u>0.77326</u>	0.07974	0.61286
Disease and Phenotype	315,186	<u>0.09125</u>	<u>0.37975</u>	0.02715	0.05215
University Admission	173	0.10982	0.83236	<u>0.11560</u>	<u>0.87283</u>
Birth Registration	232	<u>0.36206</u>	<u>0.89655</u>	0.27155	0.93103
Synthetic	803	<u>0.80946</u>	<u>0.94396</u>	0.13574	0.71481
Doremus	1782	<u>0.31481</u>	<u>0.58641</u>	0.09876	0.28226
FSDM	2015	0.07806	0.90645	<u>0.16774</u>	<u>0.95419</u>

Table B.1: Coverage Statistics DBpedia vs. ALOD Classic

The best results on the whole term and on individual tokens are underlined.

B.2 Similarity Experiments: Correlation of Narrower/Broader Overlap

Feature	ALOD XL		ALOD Classic	
	Spearman's ρ	Correlation	Spearman's ρ	Correlation
Broader Overlap Top 50	0.3733	0.2762	0.1560	0.2266
Broader Overlap Top 100	0.3873	0.2757	0.2455	0.2455
Broader Overlap Top 500	0.3855	0.2493	0.2494	0.2370
Broader Overlap Top 1,000	0.4170	0.2335	0.1782	0.1934
Narrower Overlap Top 50	0.3625	0.1485	0.3617	0.2076
Narrower Overlap Top 100	0.3986	0.1523	0.3597	0.2224
Narrower Overlap Top 500	0.5149	0.1634	0.2624	0.2134
Narrower Overlap Top 1,000	<u>0.5376</u>	0.1730	0.1928	0.1834

Table B.2: Correlation of Narrower/Broader Overlap with WS-353

The best value for Spearman's ρ is bold printed for each endpoint and each feature, the best overall value is additionally underlined.

Feature	ALOD XL		ALOD Classic	
	Spearman's ρ	Correlation	Spearman's ρ	Correlation
Broader Overlap Top 50	0.4415	0.3628	0.4112	0.4018
Broader Overlap Top 100	0.4827	0.3864	0.4614	0.4402
Broader Overlap Top 500	0.5843	0.4537	0.4102	0.4034
Broader Overlap Top 1,000	0.5960	0.4660	0.3539	0.3585
Narrower Overlap Top 50	0.3835	0.3062	0.4653	0.3983
Narrower Overlap Top 100	0.4850	0.3634	0.4706	0.4062
Narrower Overlap Top 500	<u>0.6826</u>	0.4870	0.3657	0.3513
Narrower Overlap Top 1,000	0.6698	0.5130	0.3323	0.3445

Table B.3: Correlation of Narrower/Broader Overlap with MEN

The best value for Spearman's ρ is bold printed for each endpoint and each feature, the best overall value is additionally underlined.

Feature	ALOD XL		ALOD Classic	
	Spearman's ρ	Correlation	Spearman's ρ	Correlation
Broader Overlap Top 50	0.3007	0.2021	0.3086	0.2448
Broader Overlap Top 100	0.3024	0.1888	0.3349	0.2767
Broader Overlap Top 500	0.2944	0.1611	0.2855	0.2817
Broader Overlap Top 1,000	0.3035	0.1778	0.2291	0.2302
Narrower Overlap Top 50	0.2391	0.1978	0.3360	0.2893
Narrower Overlap Top 100	0.3048	0.2249	0.3448	0.3063
Narrower Overlap Top 500	<u>0.3890</u>	0.2663	0.2012	0.1799
Narrower Overlap Top 1,000	0.3854	0.2777	0.1304	0.1232

Table B.4: Correlation of Narrower/Broader Overlap with SimLex-999

The best value for Spearman's ρ is bold printed for each endpoint and each feature, the best overall value is additionally underlined.

B.3 Similarity Experiments: Correlation of ALOD2Vec

Feature	Spearman's ρ	Correlation
ALOD2Vec Classic SG 200	0.5191	0.4900
ALOD2Vec Classic SG 500	0.5236	0.4716
ALOD2Vec Classic CBOW 200	0.5539	0.5116
ALOD2Vec Classic CBOW 500	0.5195	0.4668
ALOD2Vec XLR CBOW 200	0.5828	0.5693
ALOD2Vec XLR SG 200	<u>0.6599</u>	0.6191

Table B.5: Correlation of ALOD2Vec with WS-353

The best value for Spearman's ρ for each data set used is bold printed. The overall best score is additionally underlined.

Feature	Spearman's ρ	Correlation
ALOD2Vec Classic SG 200	0.6224	0.6089
ALOD2Vec Classic SG 500	0.6200	0.5950
ALOD2Vec Classic CBOW 200	0.6199	0.6043
ALOD2Vec Classic CBOW 500	0.6168	0.5857
ALOD2Vec XLR CBOW 200	0.6797	0.6888
ALOD2Vec XLR SG 200	<u>0.7202</u>	0.7322

Table B.6: Correlation of ALOD2Vec with MEN

The best value for Spearman's ρ for each data set used is bold printed. The overall best score is additionally underlined.

Feature	Spearman's ρ	Correlation
ALOD2Vec Classic SG 200	0.3051	0.3106
ALOD2Vec Classic SG 500	0.3173	0.3174
ALOD2Vec Classic CBOW 200	<u>0.3354</u>	0.3214
ALOD2Vec Classic CBOW 500	0.2691	0.2697
ALOD2Vec XLR CBOW 200	0.2740	0.2994
ALOD2Vec XLR CBOW 200	0.2828	0.3108

Table B.7: Correlation of ALOD2Vec with SimLex-999

The best value for Spearman's ρ for each data set used is bold printed. The overall best score is additionally underlined.

Appendix C

Further Reference Material

In the following, further reference material (examples, equations, images, enumerations) is presented.

C.1 Evaluation Measures

The following basic evaluation measures are cited according to [113, pp. 154-157]. They are not restricted to the field of information retrieval and data mining but are also used for ontology alignment evaluation.¹ The contingency matrix below (table C.1) shows which correspondences count as *true positives (tp)*, *false positives (fp)*, *false negatives (fn)*, and *true negatives (tn)*. In the literature, the matrix is also known as *confusion matrix* [112, p. 81].

	Relevant	Nonrelevant
Retrieved	true positives (tp)	false positives (fp)
Not Retrieved	false negatives (fn)	true negatives (tn)

Table C.1: Contingency Matrix

Precision (P) can be defined as follows:

$$Precision = \frac{\# \text{ of relevant items retrieved}}{\# \text{ of retrieved items}} = \frac{tp}{tp + fp} \quad (\text{C.1})$$

Recall (R) can be defined as follows:

$$Recall = \frac{\# \text{ of relevant items retrieved}}{\# \text{ of relevant items}} = \frac{tp}{tp + fn} \quad (\text{C.2})$$

F-Measure combines the two metrics and can be defined as:

$$F = \frac{1}{\alpha * \frac{1}{P} + (1 - \alpha) * \frac{1}{R}} = \frac{(\beta^2 + 1) * P * R}{\beta^2 * P + R} \text{ where } \beta^2 = \frac{1 - \alpha}{\alpha} \quad (\text{C.3})$$

where $\alpha \in [0, 1]$ and, hence, $\beta \in [0, \infty]$. A very common form is the *balanced F-Measure* (also known as F_1) where $\alpha = \frac{1}{2}$ and, thus, $\beta = 1$:

$$F_1 = \frac{2 * P * R}{P + R} \quad (\text{C.4})$$

¹For an example see the OAEI 2017 Anatomy results: <http://oaei.ontologymatching.org/2017/results/anatomy/index.html>.

C.2 Ten Challenges for Ontology Matching

In 2008, Shvaiko and Euzenat present ten challenges for ontology matching which are listed in the following in the order they are mentioned [156]:

1. Large-Scale Evaluation
2. Performance of Ontology-Matching Techniques
3. Discovering Missing Background Knowledge
4. Uncertainty in Ontology Matching
5. Matcher Selection and Self-Configuration
6. User Involvement
7. Explanation of Matching Results
8. Social and Collaborative Ontology Matching
9. Alignment Management: Infrastructure and Support
10. Reasoning with Alignments

In 2013, eight challenges were recapitulated as they were not solved yet [158]. Thereby, the following challenges were dropped:

1. Uncertainty in Ontology Matching
2. Reasoning with Alignments

C.3 Paradigmatic Relations

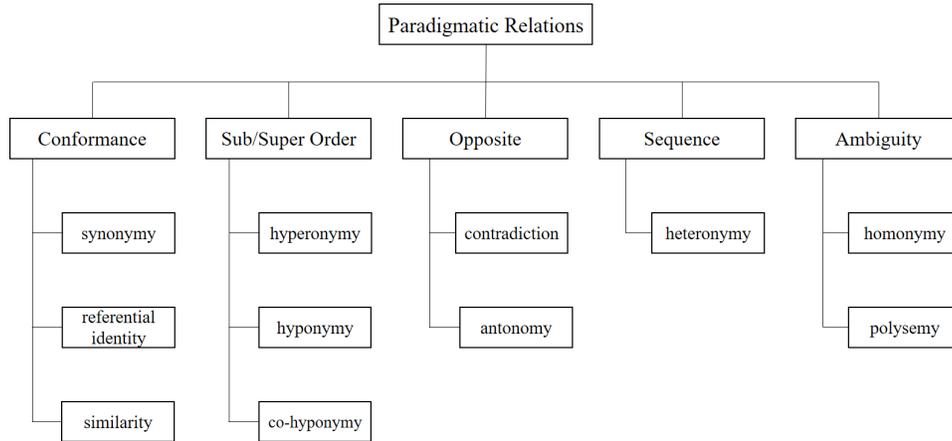


Figure C.1: Paradigmatic relations according to Busch and Stenschke [27, p. 189]

C.4 Broader Vector Space Calculation Example

For exemplary purposes, the difference between *Ernest Hemingway* and *Charles Dickens* shall be calculated. The following configuration is chosen:

- LEVEL: 2
- MINIMUM CONFIDENCE: 0.6
- LIMIT: 3
- ELEMENT BASE VALUE: FIX
- DECAY FACTOR: 0.5

In the following figures (C.2 and C.3), the confidence is given in brackets and nodes are depicted even though they do not fulfill the minimum confidence criterion. Those are dropped in the calculation. The concept *literary giant* does not have any hypernyms and is therefore a node without any outgoing edges.

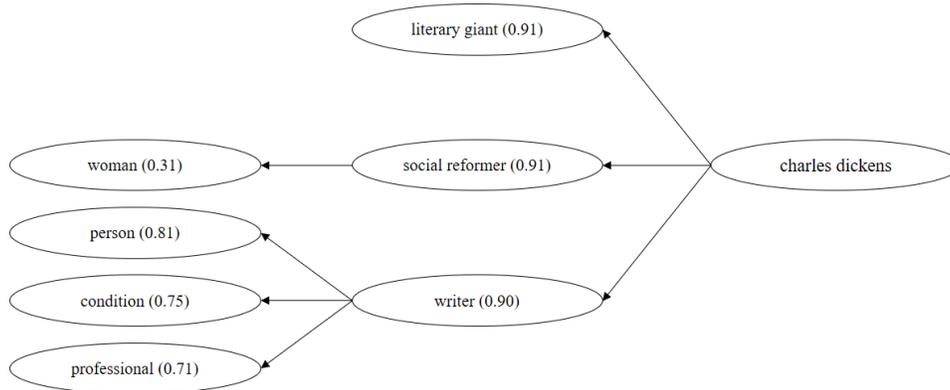


Figure C.2: ALOD Graph of the Concept *Charles Dickens*

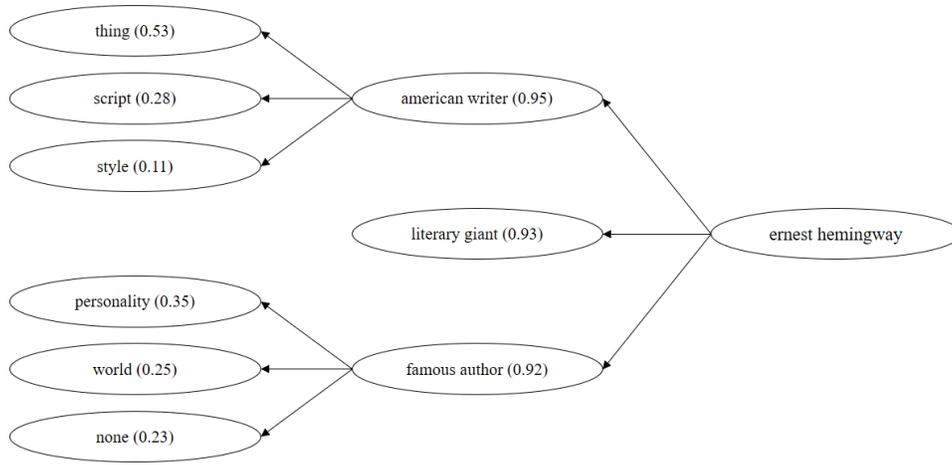


Figure C.3: ALOD Graph of the Concept *Ernest Hemingway*

The following vectors can be derived with the given configuration (see table C.4):

	ernest_hemingway	charles_dickens
literary_giant	1	1
social_reformer	1	0
writer	1	0
person	0.5	0
condition	0.5	0
professional	0.5	0
american_writer	0	1
famous_author	0	1

Table C.2: Vectors of *Charles Dickens* and *Ernest Hemingway* in Broader Vector Space

From the vectors given in table C.4, one can calculate the Euclidean distance as follows:

$$d = \sqrt{(1-1)^2 + (1-0)^2 + (1-0)^2 + (0.5-0)^2 + (0.5-0)^2 + (0.5-0)^2 + (0-1)^2 + (0-1)^2} \quad (C.5)$$

$$= \sqrt{(0.5-0)^2 + (1-0)^2 + (1-0)^2} \quad (C.6)$$

$$= \sqrt{4.75} \quad (C.7)$$

$$\approx 2.1794 \quad (C.8)$$

C.5 Levenshtein Algorithm

First published in 1965 (1966 in English) [109, p. 1], the Levenshtein edit distance is commonly used to numerically determine how similar or dissimilar two Strings are [5, p. 2355].

In its plain form, the Levenshtein distance $dist_{levenshtein}$ is the minimal number of insertions, deletions, and substitutions of characters to transfer one string into the other [5, p. 2355]. The Levenshtein distance between *child* and *children*, for example, is 3 because the first String can be transformed to the second one by applying three insertions.

$dist_{levenshtein}$ is a plain number which needs to be normalized in order to take into account different lengths of words if the scores shall be comparable. The following equation shows how to calculate the Levenshtein similarity and is given according to [30, p. 104]:

$$sim_{levenshtein}(s_1, s_2) = 1.0 - \frac{dist_{levenshtein}(s_1, s_2)}{max(|s_1|, |s_2|)} \quad (C.9)$$

C.6 Simlex-999 Instructions

Two words are *synonyms* if they have very similar meanings. Synonyms represent the same *type* or *category* of thing. Here are some examples of synonym pairs:

- *cup / mug*
- *glasses / spectacles*
- *envy / jealousy*

In practice, word pairs that are not exactly synonymous may still be very *similar*. Here are some very similar pairs - we could say they are nearly synonyms:

- *alligator / crocodile*
- *love / affection*
- *frog / toad*

In contrast, although the following word pairs are *related*, they are not very similar. The words represent entirely different types of thing:

- *car / tyre*
- *car / motorway*
- *car / crash*

In this survey, you are asked to compare word pairs and to rate how *similar* they are by moving a slider. Remember, things that are related are not necessarily similar.

If you are ever unsure, think back to the examples of synonymous pairs (*glasses / spectacles*), and consider how close the words are (or are not) to being synonymous.

There is no right answer to these questions. It is perfectly reasonable to use your intuition or gut feeling as a native English speaker, especially when you are asked to rate word pairs that you think are not similar at all.

Figure C.4: Instructions for Simlex-999 annotators [77, p. 9]

C.7 Spearman's Rank Correlation Coefficient

Under the assumption that no rank is assigned twice for one variable, Spearman's ρ is given as:

$$r_{SP} = 1 - \frac{6 \sum_{i=1}^n (R_i - R'_i)^2}{(n-1)n(n+1)} \quad (\text{C.10})$$

where R_i and R'_i are the ranks of the values of the variables x_i and y_i . [6, p. 35].

C.8 Dice Coefficient

Originally published 1945 in the context of ecology, the Dice Coefficient (originally called "coincidence index" [40, p. 298]) can be used to normalize the overlap of two sets to the [0.0, 1.0] range [30, p. 107]. The coefficient is given as follows:

$$sim_{Dice} = \frac{2 * c_{common}}{c_1 + c_2} \quad (C.11)$$

where c_{common} are common elements and c_1 are the elements in set_1 and c_2 are the elements in set_2 .

C.9 Data Model to Ontology: Transformation Rules

The conceptual and the physical data models of the SAP FSDM product were transferred into ontologies using the static rules presented in the following. The starting point was Fahad's approach [51], called *ER2OWL*, which describes how to transform an entity-relationship model into an OWL ontology. As the conceptual data model contains more semantics than the model underlying the approach, the rules were extended. The same accounts for the physical data model.

Rules Applied to the CDM

1. OWL classes are used to represent entities.
2. OWL datatype properties are used to represent attributes in the following way:
 - (a) To express that an attribute belongs to an entity, unique properties are used and membership is expressed using `rdfs:domain`.
 - (b) Primary keys can be used since OWL 2². Therefore, `owl:hasKey` is used rather than Fahad's approach of making keys functional and inverse-functional at the same time [51, p. 34].
3. OWL object properties are used to represent relations in the following way:
 - (a) The relation source is expressed using `rdfs:domain` and the target is expressed using `rdfs:range`.
 - (b) Restrictions are used to represent cardinalities of relations.
 - (c) Dependent relations are detected and treated as primary key.
4. All elements have English labels (using `rdfs:label`).
5. Inheritance relations in the conceptual data model are also covered using `rdfs:subClassOf`.

Rules Applied to the PDM

For the physical data model the same rules as above were applied when applicable with the following exceptions:

1. There are no inheritance relations.

²see https://www.w3.org/TR/owl2-new-features/#F9:_Keys

2. Foreign keys are ignored as they are semantically expressed by so called *associations* which are treated like relations.³
3. History tables that are used for versioning are ignored as they are related to the technical feature of two-dimensional versioning rather than to the ontology itself.⁴
4. Versioning attributes are ignored for the same reason.

³The reason for this is the HANA HDI CDS notation.

⁴As history tables carry the same name as their corresponding *current* table (plus some suffix), there is no problem when the ontology alignments are to be used in real mapping tasks because those mappings can be derived.

C.10 FSDM Semantic Search Server

In the following, a shortened user guide is given on how to use the FSDM Semantic Search that was written for SAP.

C.10.1 Using the Client on a Windows PC

Setup Enable Telnet (skip this step if you have used telnet before):

1. Open Control Panel
2. Open Programs
3. Select Turn Windows features on or off
4. Check the Telnet Client box
5. Click OK

Connect Given that the server is running, you can start the application out of the box. The only thing you have to do is open your terminal (Windows Key → type CMD → Enter). If the server is already running in the network you can connect to it using:

```
telnet <IP> 5500
```

If you run the server on your local machine, type:

```
telnet localhost <your specified port>
```

Search After the connection has been established, a welcome screen appears:

```
.....
.....
...SSS.....AAAA.....PPPPPP.....
..SS..SS.....AA..AA.....PP..PP...
..SS.....AA..AA.....PP..PP...
...SS.....AAA.....AAA.....PPPPPP...
...SS.....AA..AAAA..AA.....PP.....
..SS..SS..AA.....AA..PP.....
...SSS..AA.....AA..PP.....
.....
.....
-----
                               Main Menu
-----
What do you want to do?
1) Find an Entity
c) Configuration
x) EXIT
Your Input: _
```

Figure C.5: FSDM Semantic Search Welcome Screen

To get to the search enter 1 and hit enter. You can enter your search term and hit the return key. In the screenshot below the search term is “business partner”. A ranked result list with the results will appear. After your search you will be guided back to the main menu.

```
-----  
Main Menu  
-----  
What do you want to do?  
1) Find an Entity  
c) Configuration  
x) EXIT  
Your Input: 1  
  
-----  
SAP FSDM Semantic Search  
-----  
Search Term: business partner  
Your Input: business partner  
Associated Concepts:  
http://webisa.webdatacommons.org/concept/business\_partner\_  
Please hold the line. Search in progress...  
  
The following FSDM entities were found:  
1) BusinessPartner  
2) Company  
3) Organization  
4) ConsolidatedBusinessPartner  
5) IndividualPerson  
  
-----  
Main Menu  
-----  
What do you want to do?  
1) Find an Entity  
c) Configuration  
x) EXIT  
Your Input:
```

Figure C.6: FSDM Semantic Search Process

Configuration If you want more than 5 results, go to the configuration menu by typing `c` and hitting enter. You can change the number of results by entering `1` (+ return) and then your desired number of hits (+ return). After your configuration, you are brought back to the main menu.

```
-----
Main Menu
-----
What do you want to do?
1) Find an Entity
c) Configuration
x) EXIT
Your Input: c
-----
Configuration
-----
What do you want to do?
1) Set number of results to retrieve (currently: 5)
x) Back to start menu.
Your Input:
1
Set number of results to retrieve: 10
-----
Main Menu
-----
What do you want to do?
1) Find an Entity
c) Configuration
x) EXIT
Your Input: 
```

Figure C.7: FSDM Semantic Search Configuration

Exit In order to exit the program, enter `x` or `exit` and return.

```
-----  
Main Menu  
-----  
What do you want to do?  
1) Find an Entity  
c) Configuration  
x) EXIT  
Your Input: x  
Good bye!
```

Figure C.8: FSDM Semantic Search Exit

C.10.2 Running the Server (Windows, Linux)

You can run the server from the IDE or package it as JAR and run it from the command line. When using the latter option, make sure all resources are available if any are required by the selected feature; you can then start the server by going into the directory of the JAR and typing:

```
Java -jar <jar name> <port>
```

If you do not specify a port, port 5,000 will be used by default. After starting the server and after the server printed “Server ready to accept requests on port 5000” clients can start connecting to the server.

Ehrenwörtliche Erklärung

Ich versichere, dass ich die beiliegende Masterarbeit 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, August

Unterschrift