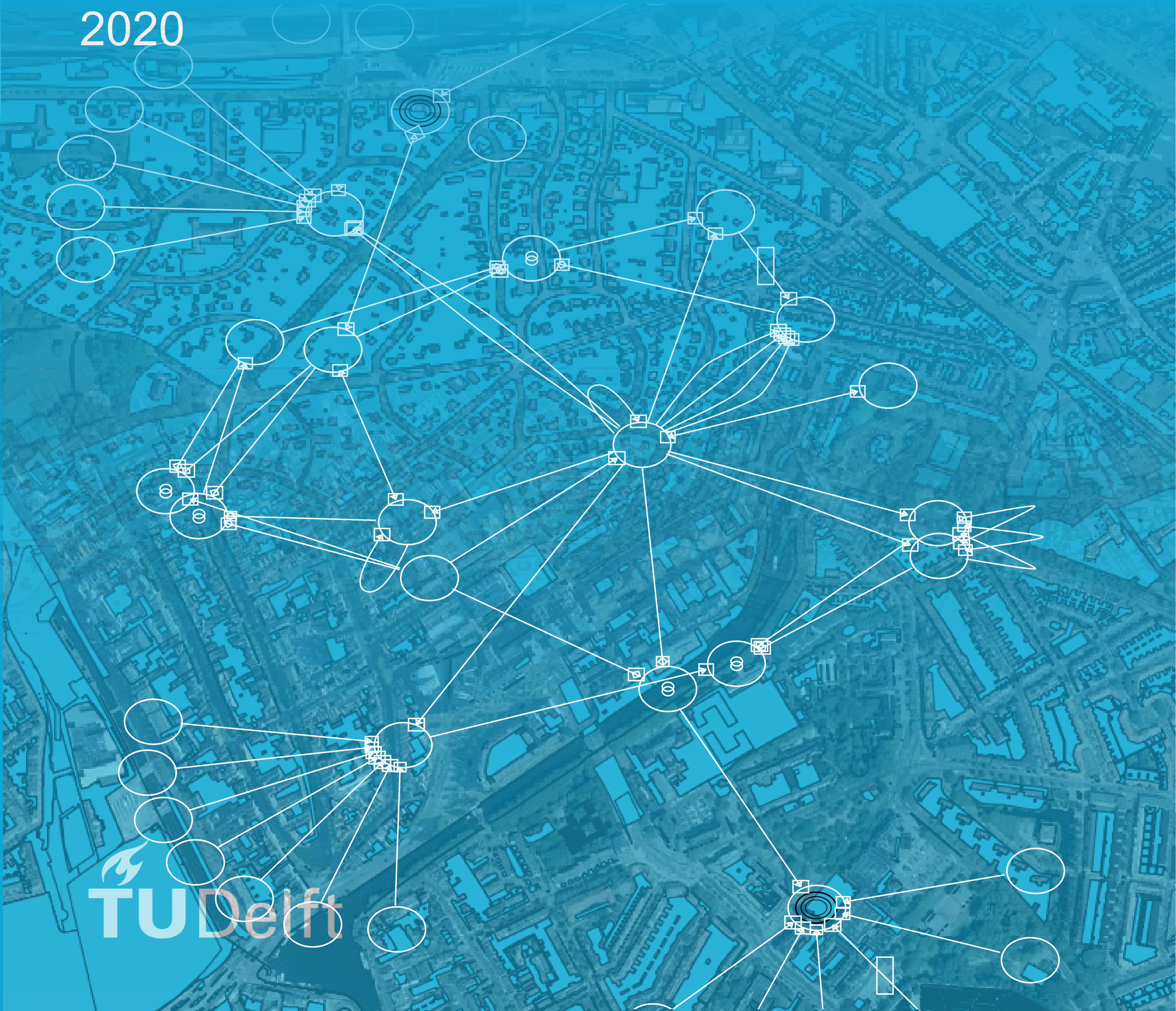
MSc thesis in Geomatics for the Built Environment

Towards the linking of geospatial government data

A study on the semantic harmonisation between data from Dutch geo-registries

Gabriella Wiersma

2020



Towards the linking of geospatial government data: A study on the semantic harmonisation between data from Dutch geo-registries

Gabriella Wiersma

March 2020

A thesis submitted to the Delft University of Technology in partial fulfillment of the requirements for the degree of Master of Science in Geomatics

Gabriella Wiersma: *Towards the linking of geospatial government data: A study on the semantic harmonisation between data from Dutch geo-registries* (2020)

The work in this thesis was supported by:

3D geoinformation group
Department of Urbanism
Faculty of the Built Environment & Architecture
Delft University of Technology

Geonovum
National Spatial Data Infrastructure (NSDI) executive committee
in the Netherlands
Amersfoort

Supervisors:	Prof.dr. Jantien Stoter
	Dr. Linda van den Brink
Co-reader:	Dr. Hugo Ledoux

Abstract

The growing complexity of many projects and applications require methods that help integrate vasts amounts of data coming from different sources. In all cases, semantic interoperability plays an important role - the meaning of content should be organized logically, as to allow machines to interpret it. The Semantic Web vision has been developing standards to support this idea, with the goal of creating a 'web of data', were pieces information can be accessed and linked to one another consistently. The ideas behind Linked Data have been explored in different areas of study. The geospatial domain presents an interesting challenge, as geospatial data offers many different views on the same physical universe. This is especially the case with geo-registries - collections of official geospatial data used by governments. This data already contains links in the form of spatial relations. However, there is no consensus on how these links should take form. Thus, this research will explore how linkable geospatial data from registries actually is by answering the following question:"To what extent can ontology-based solutions using semantic web technologies contribute to the integration and use of data from geo-spatial registries?". This question is broken down in three sub-questions that address ontology-based integration techniques (including tools and languages), the impact of dataset characteristics and the added value of semantic relations. The research includes a literature study to provide a theoretical background, a case study of two Dutch geo-registries (Basisregistratie Grootschalige Topografie and Basisregistrie Topografie) and a conceptual framework that is applied to the case study as a way to explore the main question. The results of this framework reinforce the importance of using instance-level data to understand the connections between different conceptual models and show the unavoidable subjectivity that is involved in the process - from the conceptualization of the data (model) to the creation of query rules. The findings indicate that the capabilities of ontology languages (such as OWL) are not necessarily practical for data from registries, and that custom processes might be necessary depending on the application envisioned by data users. Country-wide registries managed by different data owners and different interpretations of data acquisition rules will lead to inevitable variations in the registration of objects in datasets. And more efforts could be invested in providing quality indicators for alignments.

Acknowledgements

Several months went into writing and developing this thesis, and I would like to thank my supervisors for their continuous support throughout the process. Linda van den Brink has introduced me to the workings of geospatial information standards - which was made possible thanks to the help of Jantien Stoter. The time you both took to assist me in developing new ideas, to provide me with feedback and to help me reflect on the research were really appreciated - thank you. I would also like to express my gratitude to those involved at Geonovum for giving me the opportunity to explore these ideas. And a special thanks goes out to Hugo Ledoux for stepping in as my co-reader, and for the valuable feedback. Most of all I would like to thank Johann Meyer for his unwavering support and love, always.

Contents

List of Figures

List of Tables

1 Introduction

1.1 Background and problem statement

In The Netherlands as well as abroad there is a growing need for applications involving cross-domain data – especially considering the emergence of several web-based services in the context of open data developments. In fields related to the geospatial domain many examples are found. One could think of large construction projects, which will often use BIM (Building Information Modeling) to understand the relation between certain design choices and to ensure compliance to building regulations. Having such information represented in an open and accessible way could shed more light on dependencies - and how these translate to different situations and locations. With smart city solutions integration efforts are also a core issue, as these applications depend on data coming from vastly different sources - authorities, sensors and online networks, for example. Another example that takes advantage of data produced by users is the inclusion of crowdsourced data to enrich thematic maps made available by mapping agencies. This could help meet demands made by new users, and it could improve the completeness of data, in cases were delivering a homogeneous dataset is challenging (for example, country-wide maps maintained at a lower level). Methods are needed to correlate new information to the preexisting government data, and to expose and mediate between differences in the data quality of the sources.

In all these cases, semantic interoperability plays an important role: without it, it is not possible to interpret the content and meaning of data instances from heterogeneous sources. And since organizations often understand the meaning of concepts in different ways – depending on the domain or application for which their data is primarily collected -, conflicts are most likely to occur.

Several approaches exist for semantic data integration, among which ontology-based methods. Ontologies are used to represent a shared understanding of a certain domain and consist of a set of concepts (mainly entities and attributes), definitions and relationships. Although ontologies can take up different forms, they will always include a vocabulary with definitions of terms. Depending on the degree of formality used to create the vocabularies, ontologies can be expressed in natural language, formally defined languages or with theorems and proofs [Uschold and Gruninger, 1996]. In the context of information sciences, formal ontologies are mostly used as they give access to robust computational tools (such as inferencing and reasoning engines).

The employment of formal ontology structures for online data integration has been the subject of many studies and can be understood through the principles of the Web 3.0 – or Semantic Web. According to Berners-Lee et al. [2001], the Semantic Web extends the current Web by giving data well-defined meaning in machine-understandable language. One of the main goals is to enable systems to interpret the content of web resources, which is of great importance due to increasing volumes of data being made accessible through the web. Thus, mechanisms and applications incorporating the Semantic Web vision are being developed and standardized by the World Wide Web Consortium (W3C) in recent years.

The ideas behind the Semantic Web are being used to promote and stimulate the use of data from various knowledge domains – this is referred to as 'linked data'. Thus, linked data denotes the design principles used for accomplishing the Semantic Web vision, and can be summarized in four principles [Berners-Lee, 2019]:

1. Use URIs to name (identify) things.
2. Use HTTP URIs so that these things can be looked up (interpreted, "dereferenced").
3. Provide useful information about what a name identifies when it is looked up, using open standards such as RDF, SPARQL, etc.
4. Refer to other things using their HTTP URI-based names when publishing data on the Web.

In the geospatial domain, semantic web ideas have been explored to improve thee accessibility of data published through Spatial Data Infrastructures (SDIs). Recently, a survey managed by the European Spatial Data Research [2018] predicted that linked data will be a key factor in moving SDIs towards the next generation. Thus, governments have been investing in linked data initiatives. A recent example is the Data Integration Partnership for Australia (DIPA), which created the 'Location Spine', a model for describing the links between objects from different datasets [Car et al., 2019]. In Europe, several other initiatives can be found as well. The Italian Institute for Statistics and the Agency for Digital Italy have published public administration information as linked data. The United Kingdom Ordnance Survey has published three open data products as linked data – including the administrative geography for Great Britain. And in The Netherlands, ongoing efforts from the Dutch Cadaster have led to the publication of several (geo)registries using semantic web standards.

Publishing data online using formal ontology languages is an important first step towards semantic interoperability. However, for data to be truly linked it does not suffice to publish it with linked data standards. It must also be possible to establish meaningful connections between instances from different datasets when they relate to the same real-world objects. Although progress has been made in this area (see the overview of currently linked datasets from the Linked Open Data Cloud initiative, in Figure 1.1), a lot can still be achieved.

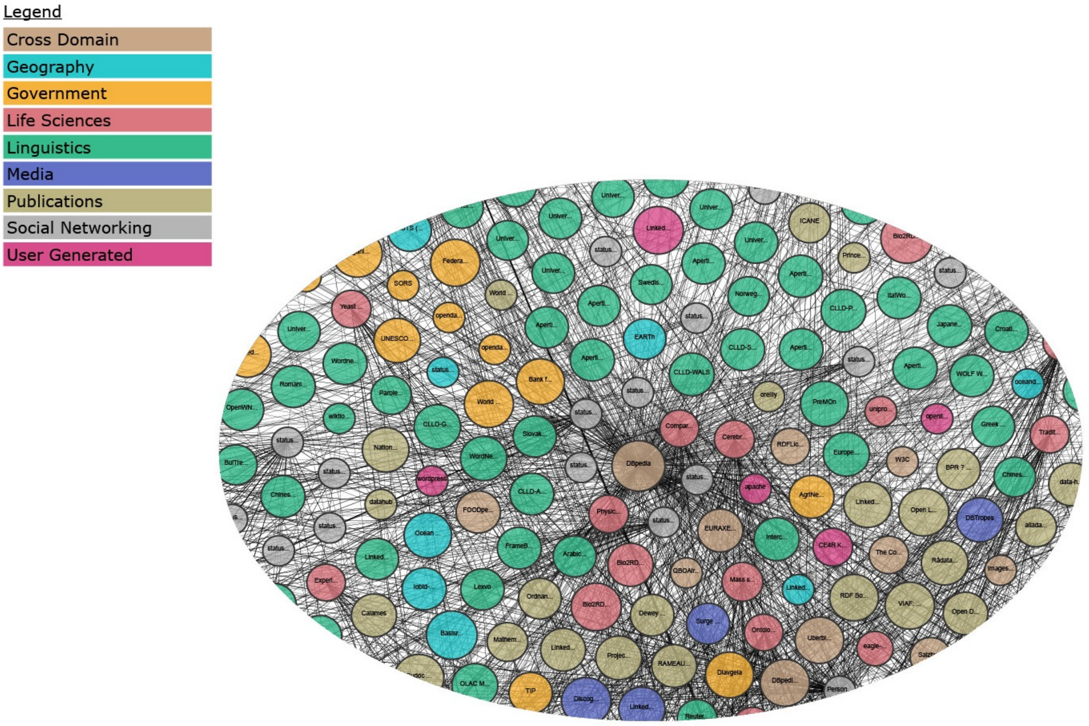


Figure 1.1: Fragment of Linked Open Data Cloud showing connections between datasets and ontologies. Adapted from The Linked Open Data Cloud [2019]

1.2 Research objectives

When speaking of geospatial data, the fourth principle of linked data can be accomplished by linking objects through their geospatial location. However, links based on spatial information alone are not always enough to communicate differences in the representation of data from heterogeneous sources. A concrete example is when the same (geospatial) instance is described differently in two sources - one may highlight the economic or cultural functions of the object, while the other focuses on the administrative viewpoint. By understanding and expressing the relationships between these different concepts used to describe instances, distinct views on reality could be handled more appropriately. Therefore, this research aims to explore the ways in which further interlinking of data semantics - through the use of semantic web technologies - leads to integration and coherence in a system of geo-registries.

1.3 Research questions

The idea of this work is to explore how linkable geospatial data published as linked data actually is. The main question of the research then becomes: "To what extent can ontology-based solutions using semantic web technologies contribute to the integration and use of data from geo-spatial registries?". To answer this question, it is important to consider the different ways in which data integration using ontologies can be accomplished. In other words, the different ways in which correspondences can be made between data model concepts - and ultimately instances - should be investigated (subquestion 1). Moreover, integration must be seen as a means to an end. Therefore, the extent to which integration is accomplished will be measured by analyzing the degree to which correspondences can be made on schema-level (subquestion 2), and by how these correspondences can help infer new and useful knowledge concerning the data (subquestion 3). Thus, the main question can be broken down into the following sub-questions:

- What type of ontology-based techniques are best suited in the case of integrating data from the geo-registries?
  - What aspects are covered in ontology-based integration frameworks?
  - Are these aspects adapted to deal with geospatial applications?
  - What types of relations and properties are often represented within the data models? How do these translate to formal ontology languages?
  - How does ontology matching relate to data interlinking?
- How does the overlap and differences between the data/models affect the correspondences between the data sources? What is the effect considering:
  - Geometric representation
  - Semantic representation
  - Granularity
- What is the added value of having custom semantic relations incorporated into ontologies?
  - Is it enough to provide correspondences based on ontology matching alone? Or should there be additional rules for data interlinking?
  - Are there advantages to the use of rule-based reasoning?

1.4 Scope of research and limitations

This research will focus on the alignment of data from geo-registries, based on the semantics of classes, properties and their values. There are many techniques available for the alignment and mapping of ontologies, and this thesis will only consider those that can be used in conjunction with semantic web technologies. The alignment itself will be performed (partially) manually. This does not entail full translation or integration of the chosen data sources. It is rather an exploratory research, which aims to investigate alignment techniques for geospatial datasets and the challenges encountered during this process. As the suitability of matchers used in the alignment depends on the structure of the input ontologies (and thus, on the modelling decisions made when creating the registries) only sources based on the ISO/TC 211 and ISO 19150-1/2 standards will be considered. While some example queries are used to explore the added benefit of ontology-based integration, the research will not investigate how to handle specific use-cases or applications - as this would require a better assessment of the data users and their needs.

1.5 Research design

The research is divided in two parts, detailed in Figure 1.2. Phase 1 consists mainly of qualitative research. It is needed to understand the different technologies involved in the semantic web and current efforts into semantic integration of heterogeneous data sources. This initial phase also involves a general analysis of the overlap between data models and instances (through visual inspection) of the selected dataset. The selection is done based on available resources and the possibility of clearly-defined semantic overlap. The output of this phase is the definition of the integration approach used to define the alignment and query framework in phase 2. The alignment framework will lead to an alignment document - and possibly reasoning rules. Their usability is then evaluated by querying data subsets. This involves a discussion of the limitations of the method, followed by the conclusions.

1.6 Reading guide

Chapter 2 addresses the background knowledge needed to develop the frameworks later on. Section 2.1 introduces some important concepts related to (geospatial) data modelling, which help understand the role that ontologies play in integration - and how other aspects are also involved in the process. In Section 2.2 integration frameworks, tools, relevant languages and data access for ontology-based solutions are discussed. This section also covers semantic web technologies and common strategies used for ontology alignment. Section 2.3 provides an overview of some projects which made use of such frameworks, and briefly explains what was done. Then, Chapter 3 presents more information on geo-registries and analyzes the case study (in Section 3.1 and Section 3.2), after which some matching algorithms for alignments are considered in Section 3.3. Chapter 4 gives a more thorough overview of the alignment framework that is used (in Section 4.1) and how/which queries can be formulated using the output of the alignment framework (in Section 4.2). The results are presented and discussed in Chapter 5. A more global discussion regarding the limitations of the methods is then given in Section 5.3. Finally, Chapter 6 contains the conclusions, recommendations and future work.

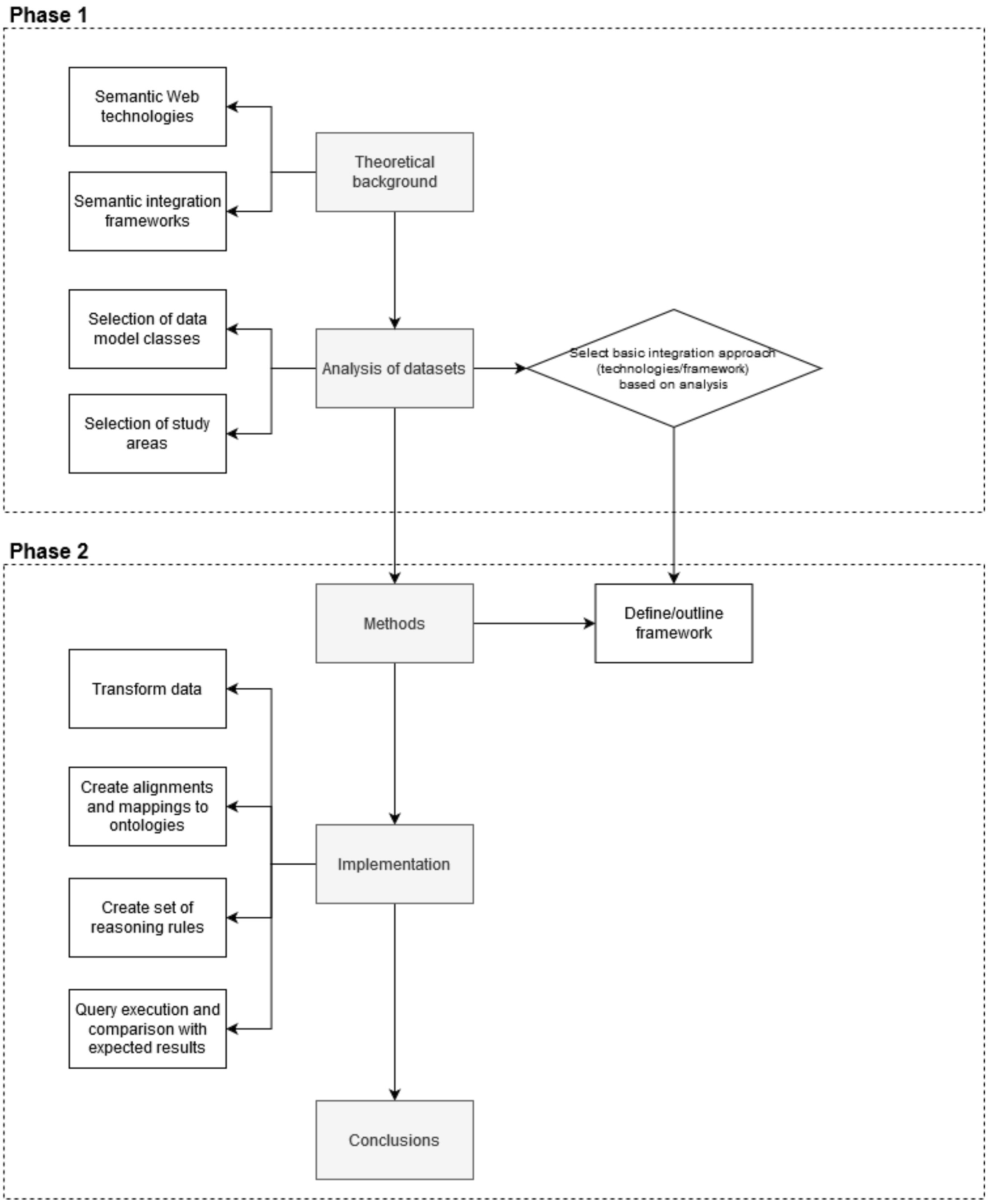

Phase 1
Semantic Web technologies
Theoretical background
Semantic integration frameworks
Selection of data model classes
Analysis of datasets
Select basic integration approach (technologies/framework) based on analysis
Selection of study areas
Phase 2
Methods
Define/outline framework
Transform data
Create alignments and mappings to ontologies
Implementation
Create set of reasoning rules
Query execution and comparison with expected results
Conclusions

Figure 1.2: Research design and phases. The main chapters of the thesis are in grey.

2 Theoretical background

In the domain of information science, ontologies provide formal description of knowledge through components such as: individuals, classes, attributes, restrictions, rules and axioms (W3C, 2004). In the context of geospatial semantics, ontology creation is either focused on modeling or encoding aspects (Kuhn [2010], Janowicz et al. [2012]). Geospatial ontology modeling is more concerned with formalizing rules and concepts as a way to constrain their interpretation (which is a design task), and is connected to the field of ontology engineering. Geospatial ontology encoding is focused on compliance to implementation standards (Kokla and Guilbert [2020]), and is related to information integration efforts. Domain ontologies (and by extension, geo-registries) are an example of the latter, as they are usually made with implementation standards in mind. Therefore, this chapter (and this research, by extension) deals with ontology encoding for the most part. Nonetheless, the following section introduces some basic notions related to geospatial modeling which are needed to understand efforts into semantic-based integration. The Semantic Web is of great relevance to this topic.

2.1 Geospatial semantic integration

Before addressing the use of semantic web frameworks for data integration, it is important to elaborate on the different aspects at play when integrating geospatial information. In other words: what does it mean to establish connections between objects that represent the same things in the physical world? The 'five universes paradigm' of Fonseca et al. [2002] offers a view on this by dividing the universe in five types for the sake of information modeling, as can be seen in 2.1.

The physical world exists independently of any perception, with the final representation of geographical features being dependent on shared agreements on a certain perception. According to this view, agreements allow connecting the physical and cognitive world - in which we distinguish between well-defined features by naming them and listing common attributes. This knowledge is then formalized in the logical layer, which is where ontologies reside. The semantic knowledge from the logical layer connects to the representation of the geographic features. In a way, the representation layer also contains formalized knowledge. However, this is related to the conceptualization of the physical space rather than a product of human cognitive processes alone. For example, the representation universe might define discrete and bounded objects based on certain attributes. These attributes can then be used to classify the object according to the logical representation. The final layer regards implementation, which deals with computational elements - such as data structures for vectors or raster and algorithms.

Thus, integration of datasets most likely entails working with different sets of agreements -

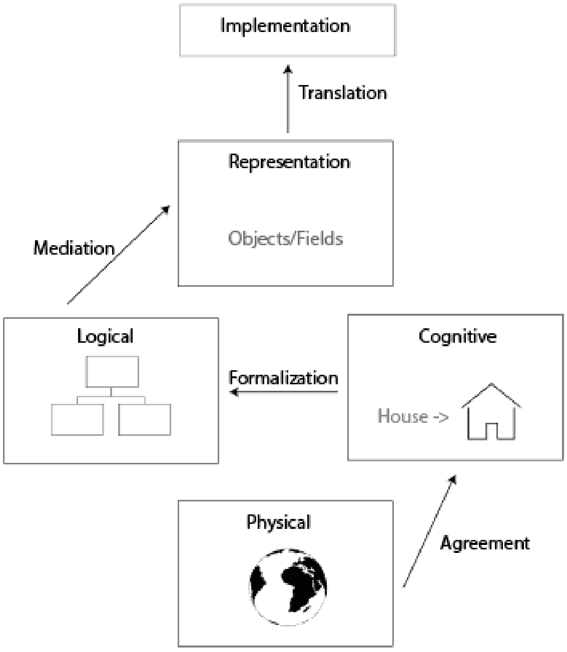


Figure 2.1: The five-universe-paradigm. Adapted from Fonseca et al. [2002]

which lead to varied cognitive, logical and representation models. Thus, geographic features (and their semantics) can be described from many viewpoints, and are context-dependent: their meaning depends on the individuals thinking about it. This is best illustrated by the meaning triangle (see Figure 2.2). The meaning triangle is an ancient philosophical concept used to understand how we interpret the world around us. It explains how symbols are used by individuals to point to a specific ’thing’. Often, these symbols (names or descriptions) are invoked by the individual to describe a concept which points to a real object. As such, symbols do not describe the reality - they describe a specific view/thought of something grounded in reality. In information systems, the idea of the meaning triangle can be found on different levels (see Sowa [2000] for a comprehensive explanation).

2.2 Frameworks for semantic integration

There are many ontology-based approaches available for integration of heterogeneous data sources. Table 2.1 provides an overview of often-used approaches or tools, according to three main aspects: the integration framework, ontology alignment and data storage/access. Section 2.2.2, 2.2.3 and 2.2.4 discuss these in more detail, after an overview of the semantic web standards that are involved is given in Section 2.2.1. For a comprehensive overview of ontology-based data integration approaches and their adoption by researchers, the work of Ekaputra et al. [2017a] can be consulted. Before going any further, some definitions should be clarified due to recurring mentions throughout this document. These definitions are based on the works of Euzenat and Shvaiko [2013], and concern the following terms:

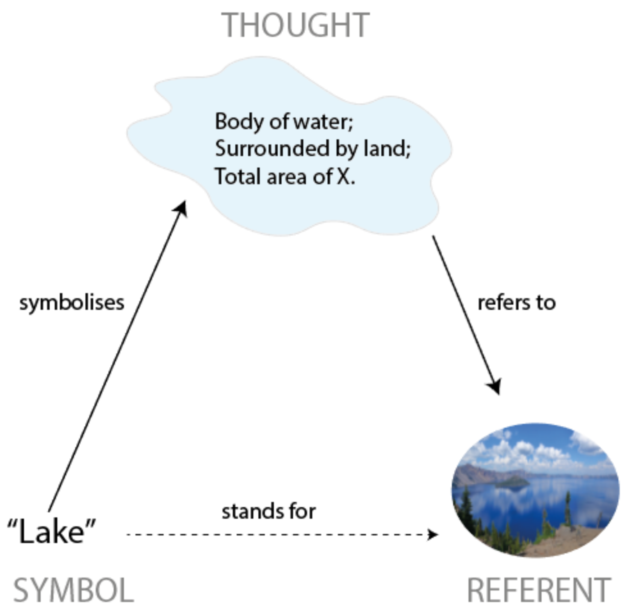


Figure 2.2: Example based on the idea of the meaning triangle.

- **Correspondences**: a correspondence expresses a relation between one or more entities (classes, properties or instances) regardless of any details related to implementations (such as specific languages or formats). In a way, geospatial instances from different sources already contain correspondences when their geometries overlap.

- **Ontology matching**: is concerned with finding relations between entities from different ontologies. There are several matching strategies that take advantage of different aspects (linguistic, structural, among others) of ontologies, and many methods can be used within these strategies.

- **Ontology alignment**: the output of the matching process is the ontology alignment. This constitutes a set of correspondences between entities of ontologies, which can be represented in many ways.

- **Ontology mapping**: the mapping is the result of interpreting the ontology alignments (or in other words, the set of correspondences) expressed in a mapping language. The mapping represents the alignments in enough detail as to allow execution for specific tasks: tools should be able to read the mappings and run them to satisfy an application need.

2.2.1 Semantic Web technologies

The Semantic Web can be represented through the Semantic Web stack 2.3, which represents the general architecture of the languages used. These languages build on one another to provide increasing expressiveness and reasoning capabilities, in a layered manner. In this

Integration framework	Single ontology Multiple ontology Hybrid ontology Global-as-view ontology	
Tools and languages	Matching techniques	Instance-based Schema-based
	Alignments and mappings	SILK SPARQL Construct Rule engines EDOAL Custom code
Data storage & access	Triplestore RDBMS File-based SPARQL endpoint Custom APIs	

Table 2.1: Main aspects of semantic integration frameworks. Based on the work of Ekaputra et al. [2017a]

research the focus will be on the stack portion related to ontologies, rules and query languages, for which well-established standards are available. These standards will be explained below in more detail, together with the structures on which they were built.

Resource Description Framework and Schema

The Resource Description Framework (RDF) forms the core on which the other semantic web languages are based. Essentially, RDF provides the user with a graph-based data model to describe entities - such as concepts or data instances. RDF structures information as triples containing a subject, predicate and object. These triples either represent entities linked through named relationships or an entity connected to a named attribute value. Although RDF is often serialized in XML, alternative notations such as RDF/JSON, Notation-3 and Turtle have become more common in recent years - as these syntaxes increase readability.

When exchanging RDF data, the relations between different classes and properties can be represented using RDF-Schema (RDFS). It standardizes which tags a user can employ for annotating resources - similar to XML or JSON Schema. With RDFS it is possible to define taxonomic relations and object relations. If a more expressive vocabulary is needed, ontologies can be used.

SPARQL Protocol and RDF Query Language

The SPARQL query language became a W3C Recommendation on March 2013. It was inspired by SQL and uses the 'select, from, where' syntax for querying and updating RDF graphs. Four types of queries can be made using the language:

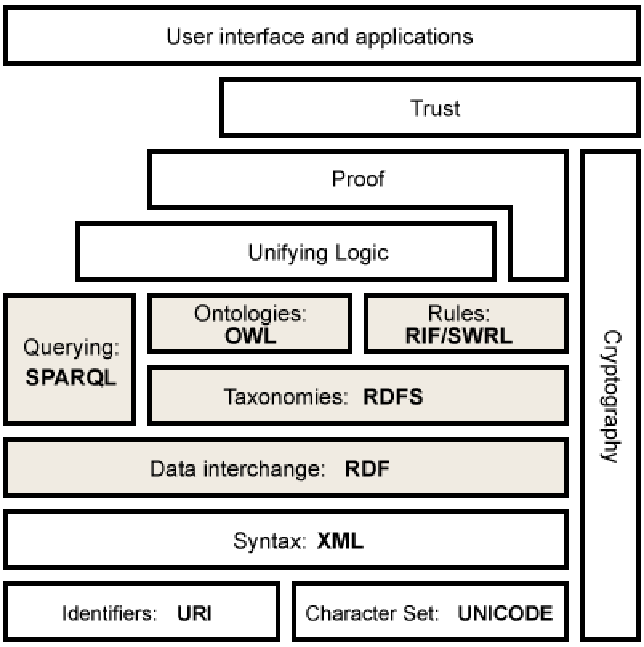


Figure 2.3: Semantic Web stack, adapted from W3C [2007]

- SELECT queries - returns variables and their respective bindings, based on the variables specified in the ’select’ clause.
- ASK queries - are used to test whether certain triple pattern can be found in the RDF graph. It always returns a boolean value.
- CONSTRUCT queries - returns an RDF graph according to a specified graph template, instead of simply returning the results values as is done with the SELECT query. Thus, the query results are used to fill the values of the graph template variables.
- SPARQL Update (SPARUL) - this is an update language with its own specification, based on SPARQL. It allows addition or removal of both triples within graphs as graphs themselves.

GEOSPARQL: a geospatial extension

To enable spatial analysis of geospatial data on the level of SPARQL, the extension GeoSPARQL was developed as an OGC standard (see OGC [2011]). It provides support for the basic OGC Simple Features functions such as sfIntersects, sfOverlaps, sfTouches, sfWithin and sfContains. It also supports some non-topological functions - although other metric-based functions - such as ’area’ - are not present in the current standard. Another spatial SPARQL extension, namely stSPARQL, does however include such functions - besides aggregate functions and support for temporal dimension. This spatiotemporal SPARQL extension was developed by Koubarakis and Kyzirakos [2010] for use in the RDF store Strabon. However, stSPARQL lacks a query rewriting mechanism and does not allow reasoning over topological relations.

Axiom type	Definition	Example (in RDF/TURTLE)	Explanation
Assertional Box (ABox)	Contains knowledge on named individuals. This includes facts on their type and property-values related to them, and (in)equality relations to other individuals.	ind:LochNess rdf:type ont:Lake	The individually defined object 'LochNess' is an instance of the concept 'Lake'.
Terminological Box (TBox)	Contains knowledge on classes defined in the ontology. This includes facts on concept subsumption and concept equivalence/disjointness.	ont:Lake rdfs:subClassOf ont:Waterbody	The concept of 'Lake' is subsumed by the concept 'Waterbody'. Thus, everything that is a lake, is also a waterbody.
Roles Box (RBox)	Contains axioms related to the properties of roles. Most commonly, they involve role inclusion, equivalence and composition.	ont:sealevel rdfs:subPropertyOf ont:waterlevel	The relation 'sealevel' is a subrole of 'waterlevel'. Thus, if an individual object has a sealevel value, it also has a waterlevel value.

Table 2.2: ABox, TBox and RBox explained

The Web Ontology Language and Rules: two paradigms

Although RDFS can be used to express many types of relations between entities, the Web Ontology Language (OWL) provides an even more expressive vocabulary. With this language, it is possible to register cardinalities of object and datatype relations and use logical operators to define the overlap of disjointedness between classes or relations. In addition to providing a comprehensive data modeling language, OWL can also be used for automatic reasoning purposes. Reasoning allows inferring implicit knowledge from explicitly modelled information.

As reasoning might require high computational power, different OWL 'profiles' have been established. Each profile corresponds to a subset of the OWL language containing trade-offs between computational power and expressiveness. To grasp the uses and limitations of these profiles, firstly the logic underpinning OWL must be understood.

Both the first and second versions have OWL variants based on description logics, which is a subset of first order logic aiming to be decidable while also remaining expressive enough. Decidability refers to inference computations being able to return results within reasonable time. Description logics (DLs) allow modelling the relationships between concepts, roles and individual names, but unlike databases they do not support close world assumptions. This means an ontology based on DL is unable to describe a situation completely. Instead, statements - also called axioms - are used to describe some partial knowledge about the situation. These axioms are often divided in three groups, described in Table 2.2. Basic axioms are often used to model (in)equality, subsumption or disjointedness, for example.

As mentioned before, the axioms of an ontology (whether related to ABox, TBox or RBox) cannot fully describe a situation, due to the open world assumption. There can be multiple 'states' that comply to a certain ontology. In other words: the absence of information does not imply the information does not exist. Thus, when trying to reason over data it is important that the ontology in question contains enough axioms that would enable logical consequences to be drawn. A logical consequence (or entailment) is an axiom that can be obtained through knowledge previously stored. This leads to an important insight regarding description logics and OWL DL specifically: it is monotonic. Adding more axioms to an ontology can only increase the amount of knowledge that is known (in the form of logical consequences). Monotonicity implies that adding knowledge to a system cannot lead to withdrawal of previously concluded knowledge.

Axioms describing more complex relations can be modelled by using and combining additional constructors. The type of constructors that are allowed to be used depend on the type of DL envisioned. One of the most expressive DLs available is SROIQ - which also forms the basis for OWL DL. All other OWL 2 profiles (OWL EL, OWL RL and OWL QL) are based on more lightweight DLs, and are aimed at different purposes:

- OWL 2 - EL is based on EL++, a subset of DLs that is optimized for reasoning in the TBox. This is most useful for complex ontologies, which define intrincate relationships between classes and properties. Reasoning tasks are optimized for inference tasks involving, for example, classification.
- OWL 2 - QL is based on DL-Lite, and is optimized for query answering involving the Abox. When the ontology is relatively simple, and large volumes of instance data are involved, this profile is more suited.
- OWL 2 - RL is based on Description Logic Programs (DLP), another subset of Description Logic related to Logic Programs. A more comprehensive description of these logics can be found in Krötzsch et al. [2014], but is out of scope for this research.

The definition of DLP and the resulting specification of the OWL 2 RL profile is especially interesting, as it indicates the need of combining rules with ontologies. The Rule Interchange Format (RIF) Working Group launched by the W3C in 2005 was an earlier example of this. Rules represent a second paradigm in knowledge modeling alongside ontologies. They can be especially useful when OWL constructs do not suffice for inferencing or when an application requires more functionalities than Description Logic is able to provide - for example, when non-monotonic reasoning is required.

According to Eiter et al. [2006] rules and ontologies can be integrated by enforcing semantic separation or semantic integration between the two paradigms. Semantic separation entails maintaining the ontology and rules layer of the semantic stack separated, while interfaces are developed between both layers to allow integration. Initiatives often involve answer-set programming (ASP), such as the approach described in Drabent et al (2009). It is also possible to integrate rules and ontologies by extending the semantics of rule languages to interact with the ontology layer. The OWL 2 RL profile and the underlying Description Logic Program are the result of this approach. However, as DLP represents the intersection between Description Logic and Horn Logic, it does not provide expressiveness beyond DL (which is precisely what is required when reasoning with rules). Therefore, the Semantic Web Rule Language (SWRL) took the opposite approach, extending OWL DL by allowing the use of Horn rules. In other words, SWRL represents the union between DL and Horn Logic, instead of the intersection. While this extends the expressivity of OWL, it also leads to undecidable cases. To overcome this, rules can be translated to OWL 2 - known as Description Logic Rules - or a decidable fragment of SWRL referred to as 'DL-safe rules' can be used.

While SWRL continues to be used in research efforts, recent recommendations such as Shapes Constraint Language (SHACL) and Sparql Inferencing Notation (SPIN) have also been explored for reasoning. SPIN was developed within TopQuadrant's TopBraid Composer - an ontology editor -, and submitted for discussion to W3C in 2011. The main idea behind this specification was to provide a mechanism that allowed storing SPARQL queries as templates, together with RDF data models and graphs. In this way, constraint checks (following closed world assumptions) and inferencing rules could be intuitively stored in an already familiar format.

```
ont:BuildingShape
    a sh:NodeShape ;
    sh:targetClass ont:Building ;
    sh:path ont:address ;
    sh:minCount 1 .
```

Listing 2.1: SHACL constraint logic: if an object is of type 'Building', it should have at least one address

```
ont:Building owl:equivalentClass [ rdf:type owl:Restriction ;
                                   owl:onProperty ont:address ;
                                   owl:minCardinality "1" ] .
```

Listing 2.2: OWL constraint logic: if an object has at least one address, it belongs to the type 'Building'.

However, in 2014 the RDF Data Shapes Working Group was formed with the intent of developing a comprehensive constraint language. Although SPIN was used as an input, the modelling language that resulted from the Working Group's effort, SHACL Core, had additional features. Essentially, SHACL was introduced as an W3C Recommendation on July 2017 as a way to supplement OWL by providing data validation functionalities. As OWL is based on the open world assumption, it is not possible to define constraints on the data to be modelled. SHACL solves this by allowing the creation of 'shapes' that express constraints on entities, such as: number of values for a property, type of value and matching patterns (see 5.1 and 2.2). A SHACL validation engine can then be used to compare input (graph) data to a document containing shapes, before the data is allowed into a triplestore, for example.

The main aim of SHACL might be data validation, however, advanced features can also facilitate the definition of rules used for more complex inferencing tasks. SHACL allows defining different types of rules, and can be used to incorporate SPARQL queries depending on the capabilities of the underlying rule engine. The specification of SHACL rules defines the framework for the language, which does also support a Javascript-based extension mechanism. Although inspired by SPIN, TopQuadrant has recognized that SHACL supersedes SPIN and is more likely to attract support in the future.

One challenge with SHACL is that it is based on the closed world assumption. At first sight, this might allow to derive more facts from the data since there is no undecidability as with OWL DL. However, it neglects the fact that most datasets only represent a partial or context-dependent view of the reality. OWL DL-based reasoning is most useful for representing structured knowledge (and relations between classes and properties), while SWRL rule-based reasoning is better suited for deductive knowledge - useful when answering queries, for example.

2.2.2 Integration framework

According to Wache et al. [2001] there are three main architectures used in ontology-based integration: the single, multiple and hybrid approach (see 2.4). The single-ontology approach consists of integrating all data sources based on one global ontology. This is especially useful when the sources provide the same views on a domain, with the global ontology acting as a 'common ground' translation that allows interpreting the information in all sources through one vocabulary. The main disadvantage of this method is that changes in one of the data sources might require updating the global ontology - which affects its relation with all sources. In the multiple ontology approach, each data source has its own local ontology, and relations are established directly between all the sources. In this way, changes in one source do not necessarily affect other sources, making maintenance of mappings easier. However, defining these mappings between all related concepts is a time-consuming process, and without a shared vocabulary the comparison between ontologies becomes a complex task.

With this in mind, hybrid approaches were created. These still allow meaningful mappings to be created directly between local ontologies, but a shared vocabulary can be used to make comparisons easier. This vocabulary defines basic terms used by all data sources - it can also take the form of an ontology. Ekaputra et al. [2017b] came up with an alternative to multiple ontology approaches, called 'Global-as-View'. The difference is that mappings are defined between each data source and the shared vocabulary (now a global ontology). Thus, there is no need to change local ontologies to fit the shared vocabulary. This approach is useful when the number of data sources to integrate is high, or if it is expected many sources will have to be added to the framework eventually.

2.2.3 Languages, techniques and tools

Regardless of the approach chosen for integrating geo-ontologies, a key issue remains: which is finding correspondences between entities (be it concepts or instances) from different data sources. The problem involves creating mechanisms for finding the correspondences (ontology matching), as well as deciding on how to express the results (the alignment, or the set of correspondences between two ontologies) in a machine-readable way (ontology mapping). The general workflow for this process is illustrated in Figure 2.5.

Matching techniques

Throughout the years many matching techniques have been developed, commonly classified according to 2.6. The ideas behind the classification are explained in Euzenat and Shvaiko [2013]. Below, a short description of the concrete techniques is given.

- Formal/Informal resource-based: these techniques use other established resources to improve the matching process. Formal resources could be global/domain ontologies, for example.

- String-based and language-based: both techniques make use of word similarity for concepts of the ontologies. String-based methods are far more simple as they make use

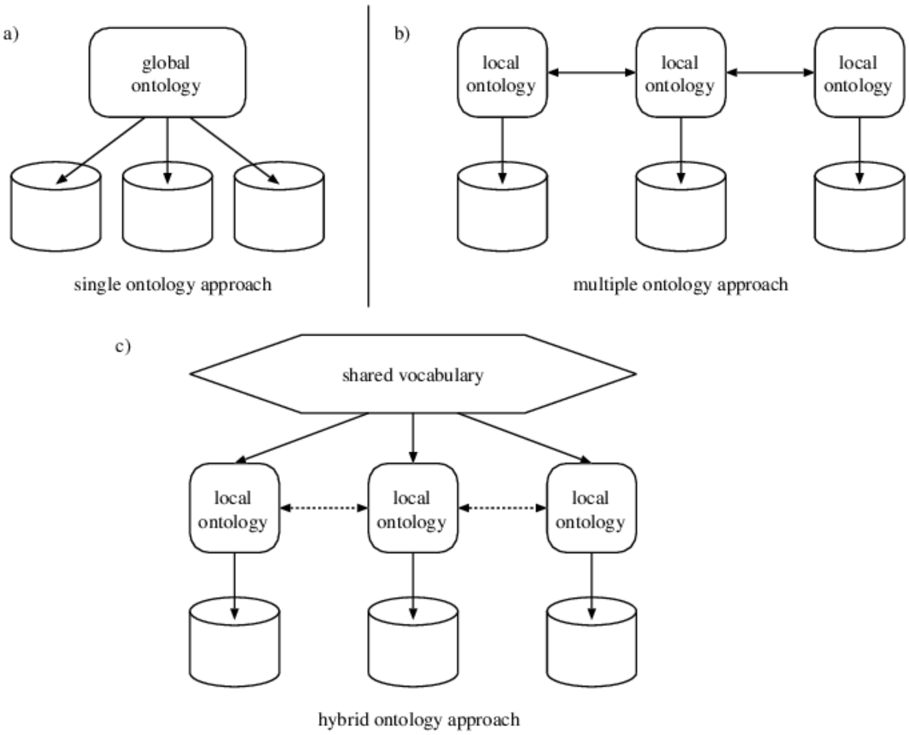


Figure 2.4: Basic approaches towards ontology-based data integration, from Wache et al. [2001]

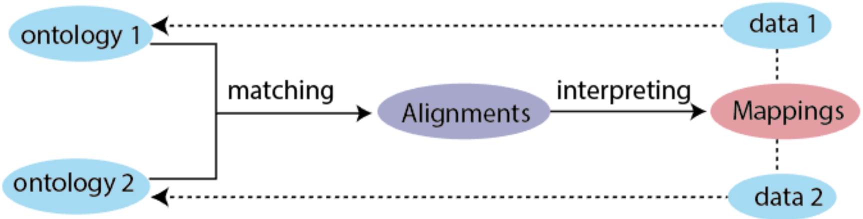


Figure 2.5: Relations between ontology matching, alignment and mapping, adapted from Euzenat and Shvaiko [2013]

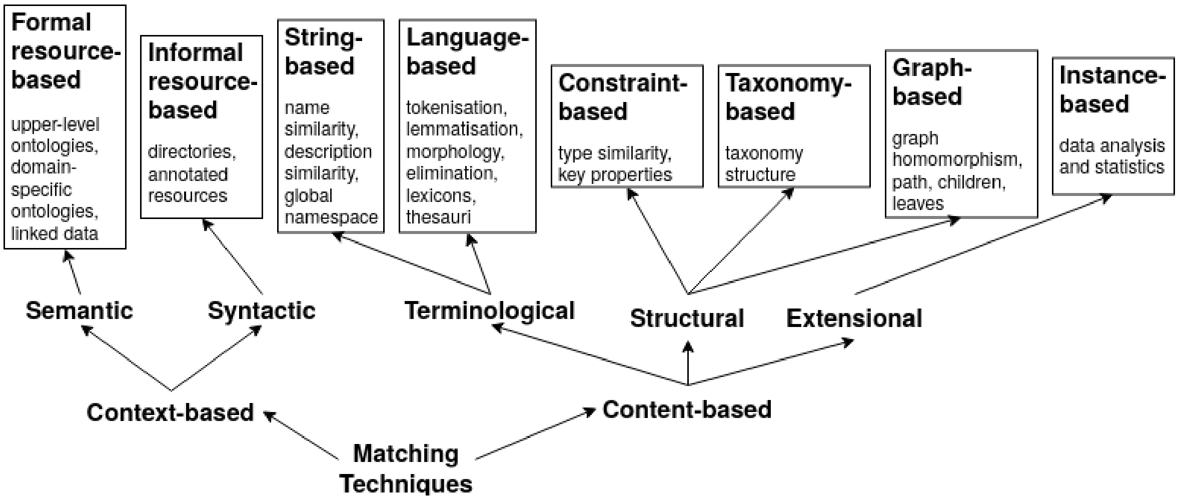

Formal resource-based
upper-level ontologies, domain-specific ontologies, linked data
Informal resource-based
directories, annotated resources
String-based
name similarity, description similarity, global namespace
Language-based
tokenisation, lemmatisation, morphology, elimination, lexicons, thesauri
Constraint-based
type similarity, key properties
Taxonomy-based
taxonomy structure
Graph-based
graph homomorphism, path, children, leaves
Instance-based
data analysis and statistics
Semantic
Syntactic
Terminological
Structural
Extensional
Context-based
Content-based
Matching Techniques

Figure 2.6: Classification of main ontology matching approaches, adapted from Euzenat and Shvaiko [2013]

of the structure of words or annotations - often involving string equality or sub-string tests, for example. However, language-based methods are more reliable if homonyms or synonyms are used. These methods make use of natural language processing (NLP) and can take the context of words into consideration. A popular approach is that of word embeddings: words are mapped to vectors, based on trained models. The distance between these vectors indicates the similarity between words. One limitation of such methods is that the models used to translate vectors must be properly trained - for domain-specific words or expressions this might be more difficult.

- Constraint-based: considers structural aspects of ontology entities - such as the domain of properties, or the attribute (value) types -, to calculate similarity.
- Taxonomy-based and graph-based: essentially, both methods use the structural properties of ontologies to improve mappings. The main assumption is that entities also share similarities with their neighbours. Taxonomies expose hierarchical relationships, so this method uses information on subsumption relations to show that two similar entities between ontologies are also related to similar entities within the ontologies. Graph-based techniques are more general: ontologies are considered as labelled graphs, and if two nodes between ontologies are similar according to some metric, then this propagates through their neighbour nodes.
- Instance-based: the individual instances of the ontology are used to determine the similarity between concepts. The main idea is that classes that share the same individuals are also similar in some way. Common techniques involve the use of Bayes classifiers, formal concept analysis or other statistical approaches. More simple methods are also common, including the use of hamming distance or Jaccard similarity.

Alignment and mapping

The correspondences between entities in ontologies (be it Abox instances or Tbox concepts) have to be expressed somehow - using mapping languages. Kovalenko and Euzenat [2016] divide these languages in two main types: declarative and procedural. The first type expresses things independently from how they are processed. Examples of declarative languages are: OWL, SKOS and EDOAL. Procedural languages express the processing as well - they can be used for specific applications. Examples of procedural languages are SPARQL Construct, SPIN/SHACL and Jena rules. Below these languages are discussed:

Create overview table with pros/cons of the mapping languages mentioned below

- **OWL**: Relations of equivalence and subsumption can easily be modelled in OWL, and one-to-one correspondences can be expressed using owl:sameAs, owl;equivalentClass and owl:equivalentProperty for instances, classes and properties respectively. However, Halpin et al. [2010] found that these constructs are often used in different ways by the linked data community. This signalizes one of their main limitations: some things are very similar under specific circumstances, without being perfectly equivalent. Also, as OWL's main function is knowledge representation this language is not suited for data transformations between ontologies. As an example: a property 'name' in a source ontology has to be mapped to the properties 'first name' and 'last name' in a target ontology by mapping the first string of 'name' to 'first name' and the second string to 'last name' - this is not possible with OWL alone.

- **SKOS**: the Simple Knowledge Organization System (SKOS) is a RDF vocabulary for expressing relationships across thesauri or lightweight ontologies, through the use of skos:Concepts. Correspondences between concepts can be expressed through the following relations:

  - **skos:exactMatch**: Transitive property that donates equivalence between two concepts. It will most likely not be used, as equivalence cannot be found when using approximate matching (which is the case with extensional matching).

  - **skos:closeMatch**:A non-transitive property that links two concepts that are similar enough to be used in some applications. The extend to which they are similar should be annotated somewhere, naturally.

  - **skos:relatedMatch**: is symmetric (although this can be changed when making a sub-property) and can be used to create associative mappings between concepts. These are mappings that express relatedness more than similarity.

  - **skos:broadMatch/narrowMatch**: sub-property of SKOS:broader/SKOS:narrower, specifically made for expressing alignments. Means a concept is more general or specialized than another. However, using this in cases were a word in one model subsumes another (identical) word in a second model might be confusing.

  Again, such mapping vocabulary does not allow to express more complex correspondences - only the relation between two concepts can be represented. And although SKOS helps represent similar and associative relations (when concepts are not necessarily similar but related somehow), it is not possible to register the extent to which these assertions hold.

- **Alignment format**: is a general format for representing alignments (it is not based on any particular language), made with the intention of providing automated matching tools with a standard output format. It can be handled through the Alignment API [1], which is the official format used during OAEI evaluations. While the alignment format is classified as a declarative language, the Alignment API can actually be used to render some types of alignments in (executable) languages. Each alignment document contains a set of correspondences made between two matched ontologies. Each correspondence has the following elements:

  - *entity1*: the first entity that has been matched
  - *entity2*: the second entity that has been matched
  - *relation*: the relation that holds between both entities. This could be one of the default relations (equivalence, subsumption, disjointedness or overlap) or it could be a custom relation.
  - *strength*: denotes the confidence that the correspondence is correct. There is no standard way to calculate this value, so it depends on the matching process on itself.
  - *id*: which identifies the correspondence

  One of the advantages of the alignment format is that it provides different levels of expressiveness, depending on the type of correspondences that are needed for an alignment. There are three main levels of expressiveness: 0, 1 and 2. Level 0 is used for correspondences between entities identified by URIs. Level 1 was intended for correspondences between two sets/lists of entities identified by URIs - according to API documentation [2] this has never been used. Finally, level 2 alignments can describe more complex correspondences with the use of expressions or formulas. The Expressive and Declarative Ontology Alignment Language (EDOAL) is an example of a level 2 alignment format language. EDOAL can be used to apply restrictions on matched entities - to narrow the scope of the match. This is useful if, for example, a class needs to have specific property values in order to match with another class. EDOAL also allows transformations between property values, and provides 'linkkeys' that can be used to express the context under which aligned entities are equivalent. EDOAL can be rendered in RDF, OWL or SPARQL. However, not all EDOAL alignments can be translated to these languages ([Euzenat, 2018]).

- **SPARQL CONSTRUCT**: Depending on the application, it might more useful to have executable mappings. Such mappings contain an interpretation of correspondences, which can be used 'on the fly' in applications. SPARQL CONSTRUCT can be used as such, as they generate RDF graphs based on a input graph. They can be used to define mappings by describing how patterns in a graph can be translated to another

[1] http://alignapi.gforge.inria.fr/
[2] http://alignapi.gforge.inria.fr/format.html

graph. One thing to keep in mind is that the outcome of the CONSTRUCT query might be affected by the reasoning capabilities of the involved SPARQL endpoints. If the matching pattern in the CONSTRUCT query is not explicitly found in the initial RDF graph and no reasoner is available for that graph, then the returned graph might be incomplete or non-existent. Nonetheless, SPARQL CONSTRUCT queries are very useful when more complex transformations are needed between ontologies - the use of the SPARQL language might be more intuitive as well.

- **SHACL**: can be used to create executable mappings by means of rules. The SHACL Advanced Features specification [3] defines SHACL rules as an extension to SHACL shapes. Shapes are sets of constraints that can be applied to nodes (ontology classes, properties or instances). Rules follow the same principles, but generate inferred triples instead of validation reports. SHACL knows two types of rules: triple rules and SPARQL CONSTRUCT rules.

2.2.4 Data storage and access

There are two main storage options for RDF data: triple stores or in-memory stores. When dealing with prototypes or relatively small amounts of data, in-memory storage is often used. The Protege OWL API makes use of this. As opposed to traditional database approaches, an in-memory database keeps the data in main memory. This usually leads to faster performance at the cost of persistence - data might be lost when restarting. Triple stores are graph databases specifically built for the retrieval of RDF fragments, through the use of query languages. Although these triple stores are optimized for RDF data, when large volumes of data are involves some might opt to use alternative methods. One example is the ontology-based data access method which allows accessing virtual RDF graphs while the underlying sources are represented in relational databases, as explained in Ekaputra et al. [2017b]. Some triple stores also offer in-memory storage, although most of them use on-disk solutions.

In order to handle the storage and querying of geospatial objects, triple stores need to fulfil additional requirements. To determine if geospatially enabled triplestores provided satisfactory query performance and if they complied to the GEOSPARQL specifications, Huang et al. [2019] conducted a assessment. The researcher also considered the support of rule-based reasoning, and analyzed the following RDF stores: RDF4J, Jena TDB+GEOSPARQL, Virtuoso, Stardog and GraphDB. While GraphDB outperformed others at non-topological queries, the open-source solutions RDF4J and Jena were generally better at spatial search queries. The advantage of these frameworks is that they allow more freedom in defining functions and rules. For example, Stardog only supports rule-based reasoning under certain conditions - and GEOSPARQL functions are only partially available for use. And while Virtuoso supports SPIN rules and allows creating custom rule functions (also refered to as 'magic properties), this feature is only available in their enterprise edition. Thus, for proof-of-concept projects it might be more reasonable to create the necessary functionalities from scratch, instead of working around the limitations of commercially available software.

Concerning the access of RDF data, either SPARQL endpoints or custom APIs can be developed.

[3] https://www.w3.org/TR/shacl-af/

2.3 Related work

In the 1990s, the concept of ontologies was introduced into the field of geographic information science (Winter, 2001, as cited in Sun et al., 2019). In order to understand the development and relationship between ontologies, Uschold [2000] classified them into global and local. Global ontologies represent a common and shared agreement between different groups or units and can be found at various levels of hierarchy – on the level of an organization as well as a domain. Local ontologies are then characterized by the fact that they have to be aligned to a global ontology, as they are not a primary source of reference [Janowicz, 2012]. Although the distinction between global and local ontologies can be fluid, this classification will be used to understand research efforts in the development of geo-ontologies.

Although global ontologies can be found within organisations and at smaller scales, the idea of creating one top-level or universally agreed upon ontology for the geo-information community has attracted some interest from researchers. Kuhn [2003] relates to this idea by introducing the concept of semantic reference systems. Similar to how spatial reference systems enable integration of spatial data across systems, semantic systems should allow integrating data cross heterogeneous semantics. A semantic reference system has a semantic reference frame and a semantic datum. The semantic datum allows projecting data models to simpler representations and translating data from different models. The reference frame consists of a conceptualization of a certain universe of discourse and can be formalized by using ontologies. To realize a semantic reference frame (and ultimately the system), the first step would be to extract semantic primitives and formalize their meaning. An experimental implementation of such a system was realized in Baglatzi and Kuhn [2013] by creating an ontology for the land cover domain based on the Conceptual Space Markup Language (CSML).

More recently, Sun et al. [2019] attempted to create a comprehensive ontological framework named GeoDataOnt, which they believe could provide a standardized representation for semantic geo-information. Their research involved finding the main semantic issues related to geospatial data sharing and integration. The results were then used to identify relevant top-level terms based on essential, morphologic and provenance characteristics of geospatial data. Each type of characteristic was formalized in an ontology, represented in OWL. However, as the quality of framework has not been evaluated, there is no insight into whether the entities defined in the ontologies (classes, properties, relations and instances) are able to accommodate views from different local geo-ontologies. Furthermore, the ontologies were created manually. This method is not the most suited for such complex tasks, as it is error-prone and turns updating into a labour-intensive task.

While data integration might motivate the use of semantic web technologies, it is not an end in itself. Often, integration represents a pre-requisite to accomplish more advanced tasks. Therefore, another approach is to focus on the development and interlinking of local geo-ontologies, which are used to facilitate more specific tasks. Zhang et al. [2010], for example, explored the use of semantic web technologies in the context of disaster and emergency management. Their research focused on finding a solution for searching feature level geospatial data based on their content – instead of traditional metadata keyword searches -, by means of OWL ontologies. A more recent example is the research of Chen et al. [2018], which introduced a mapping mechanism and a semantic translation engine to generate domain ontologies that can be used for the computation of urban density indicators. They use OWL-DL to express their ontology. Wang et al. [2018] also developed an ontology-driven integration system that allows exploring information related to geology and palaeontology, with the goal of improving the compatibility between local and global geologic standards. Futia et al. [2017] used SPARQL queries to investigate inconsistencies in procurement data, and found problems related to incoherent payments under ongoing contract and multiple registered business names. Their research was motivated by developments in the publication of Open Government Data (OGD). More recently, the research of Homburg and Boochs [2019] emphasized the importance of data quality in geospatial linked data. The authors indicate that data quality could be used by reasoners for decision making processes – to help assess the reliability of the information contained in the employed data sources. To this end, data quality requirement profiles were created, which define metrics and value ranges indicating the reliability of the data regarding a certain use case. The profiles were then converted into SWRL[4] reasoning rules. These rules are applied to the data stored in a triplestore, and through GeoSPARQL [5] queries it is possible to find out if the available data is suitable for the intended purpose.

The integration between local ontologies from government authorities has also attracted considerable attention. Years ago, Alani et al. [2007] were already exploring the benefits of using semantic web technology to enable better re-use of public sector information in the UK. The researchers collected data from several public sector organisations and designed OWL-DL ontologies for each dataset. Mappings were created between both concepts and instances of the datasets using CROSI, an alignment tool offering a wide range of mapping algorithms. Mapping on instance level was done by creating scripts searching for duplicates of specific instances, which were then connected through owl:sameAs links. Finally, all local ontologies were manually mapped to the best matched terms in the government's reference taxonomy (the Integrated Public Sector Vocabulary, or IPSV). This integration of data sources provided the researches with insights into the quality of datasets. For example, joining business information from different sources on their address coordinates revealed mismatching information.

Yu et al. [2017] explored the use of ontologies to avoid data duplication between Australian governmental authorities. The researchers build ontologies for Points of Interest datasets from different organizations. Then, the data integration tool Karma was used to convert the source data to RDF format. Finally, automated reasoning (through SWRL rules based on geometry, topology and policy rules) was proposed as a solution for finding the best location in the context of emergency response applications. The methodology was limited to handling point geometries.

In The Netherlands, Brink [2018] has explored cross-domain semantic harmonisation between different domain models within the Dutch SDI - starting from the Information model Geography. The method used in the research was based on manual matching, as the focus was to promote better data re-use by involving stakeholders personally. Firstly, the semantic overlap between models was found and published in a register using a domain independent classification. The visualisation of the register then exposed the semantic conflicts, which were discussed with the model-owners. Further efforts by the Dutch National Mapping

[4] https://www.w3.org/Submission/SWRL

[5] https://www.opengeospatial.org/standards/geosparql

Agency, Kadaster, have led to the publication and linking between three base registries containing information on: buildings and addresses (BAG), topography (BRT) and cadastral parcels (BRK). The research of Ronzhin et al. [2019] describes the process of building a knowledge graph for these and other official datasets. Most of the data was aligned by using spatial relations, by means of topological analyses based on GeoSPARQL queries.

Regardless of the approach chosen for integrating geo-ontologies, establishing correspondences between entities from different data sources remains a key research issue. The problem involves creating mechanisms for finding the correspondences (ontology matching), as well as deciding on how to express the results (the alignment) in a machine-processable way. To this end, many tools have been created and made available online. In order to evaluate the performance of such tools, the Ontology Alignment Evaluation Initiative (OAEI) [6] organizes annual contests. The initiative publishes benchmark datasets from different domains, composed of two ontologies and a reference alignment developed manually by experts. The performance of the matching tools is measured by comparing the results of their alignments to the reference alignment. However, many of the alignment algorithms used by general matching tools do not account for the spatial characteristics of geo-information. Moreover, the OAEI does not provide benchmark datasets that are representative of the geospatial domain.

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

3 Geo-registries and matching

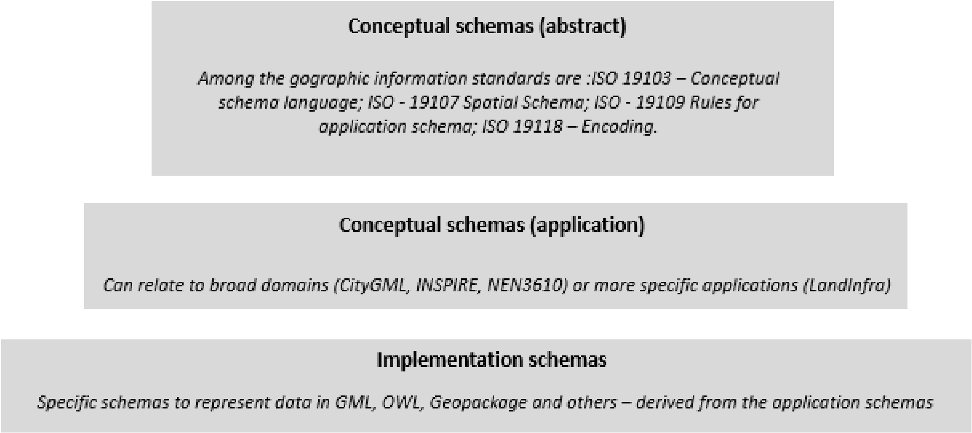
In Chapter 2 a general overview was given of the main types of matching strategies. Each strategy type can be implemented in many different ways, using (variations of) methods and algorithms developed for matching - this is referred to as a 'matcher'. Often, multiple matchers are combined when performing alignments as this can improve results. However, it is out of the scope of this research to review all the different options of matchers available for each strategy. Therefore, Section 3.2.1 introduces the main characteristics of geo-registries by means of a case study and uses this information to narrow down the matching strategies that will be considered. Then, a description of the main methods related to each strategy is given in Section 3.3. In the next chapter, the chosen methods will be discussed and fine-tuned to the characteristics of the case study data.

3.1 Geo-registries and linked data

The term geo-registries refers to base registries containing geospatial information. In the context of governmental initiatives, base registries can be understood as collections of reliable information that can or must be re-used for the provision of public services. Public administrations or appointed organizations are in charge of the use, updating and preservation of the information contained in these registries European Commission [2017]. Examples of themes commonly handled in geo-registries are: Administrative units, Addresses, Buildings, Cadastral parcels, Transport networks, Hydrography, and Land use European Commission [2020]. Such registries are a key component of Spatial Data Infrastructures (SDIs). If available to the public, the data (and models) can usually be found through geo-portals.

Linked (open) data can be used to encourage transparency in the public sector and facilitate data re-use. By allowing similar or identical information fragments to be linked explicitly and retrieved at once, findability and access to data can be improved. While initially the main aim of SDIs was to support data sharing between public bodies, there is currently a shift towards greater involvement of the private sector and citizens as key stakeholders Vancauwenberghe and van Loenen [2017]. This is related to the notion that the successfulness of open data systems depends on the extent to which the data is being used van Loenen [2018]. Higher data usage translates into higher economic or social value of the data. And in order to enable constant data circulation, users need to have access to reusable and interoperable data - hence the importance of exploring linked data.

Although developers and owners of such registries can follow their own specifications and conventions, the ISO/TC 211 produces the *de jure* standards for geospatial interoperability between domains and across organisations - and such standards are one of the key components of Spatial Data Infrastructures Rajabifard et al. [2002]. These ISO standards provide, among



Conceptual schemas (abstract)

Among the gographic information standards are :ISO 19103 – Conceptual schema language; ISO - 19107 Spatial Schema; ISO - 19109 Rules for application schema; ISO 19118 – Encoding.

Conceptual schemas (application)

Can relate to broad domains (CityGML, INSPIRE, NEN3610) or more specific applications (LandInfra)

Implementation schemas

Specific schemas to represent data in GML, OWL, Geopackage and others – derived from the application schemas

Figure 3.1: Layers of standards: from conceptual schemas to implementation. Adapted from Jetlund et al. (2019)

other things, abstract schemas to be used for information modelling. Based on these abstract schemas, application schemas can be derived to address more specific needs (see Figure 3.1. An application schema is defined as a “conceptual schema for data required by one or more applications” (ISO 19109). Geo-registries are the result of this process: their structure is based on conceptual schemas, and the data can be accessed based on implementation schemas.

As geo-registries are concrete application schemas of conceptual schemas, using a case study can improve the understanding of what defines these datasets. Notably, The Netherlands already have a production Linked Data platform at PDOK [1], which is currently being used to publish and maintain several geospatial datasets that have been implemented in OWL – including the Key Registers for buildings and addresses, cadastral parcels, and topography. Given that these Dutch initiatives are in line with aforementioned standards and publicly accessible, they will be used in the case study to further guide the selection of the matching strategies and specific methods.

3.2 The case study

In the following section the datasets for the case study will be introduced. As mentioned in the beginning of this chapter, due to the large number of matchers created for ontology alignments it is necessary to first establish which matching strategies can and should be used. This is done by examining: the background and general information on the datasets, the content of their information models and their linked data representations. The conclusions, given in Section 3.2.5, will be used towards answering one of the main questions of this research. The main methods used in the chosen strategies will be discussed in Section 3.3, and used for the actual alignment described in Chapter 4.

[1] https://www.pdok.nl/linkeddata

3.2.1 Datasets

To answer the research questions a subset of data from two geo-registries in The Netherlands will be used. Much work has been done on creating a Dutch system of registries based on standardized information models. Nonetheless, more research is needed to determine how data from different models can be better combined and to what extend integration can be accomplished. Therefore, the Basisregistratie Grootschalige Topografie (BGT) and the Basisregistratie Topografie (BRT - Top10NL) are the subject of this study. These datasets have been selected for several reasons. Firstly, there is a significantly high semantic overlap in their domains, meaning there is a high potential for integration (see for comparison between both datasets the fragments shown in Figure 3.2 and Figure 3.3. Secondly, BGT contains more detailed information and could be used to automatically extract the Top10NL in the future. Knowledge on the semantic links between their data objects could provide valuable insights. Finally, there are currently no efforts into integrating both models by using semantic web technologies - as the BGT has not yet been officially published using linked data standards.

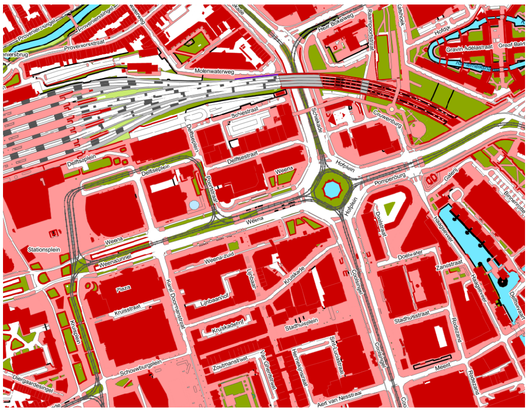


Figure 3.2: Example of BGT data from Rotterdam

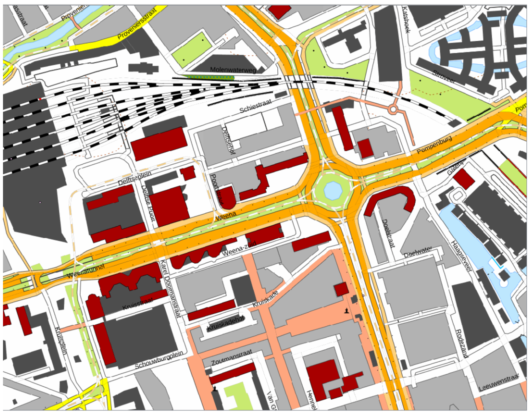


Figure 3.3: Example of Top10NL data from Rotterdam

3.2.2 Basisregistratie Grootschalige Topografie

The Basisregistratie Grootschalige Topografie (BGT) was developed to provide a uniform representation of the large scale topography of The Netherlands. It covers a scale between 1:500 and 1:5000, and contains information on topographic objects such as buildings, roads, watercourses and terrains (van Infrastructuur en Milieu [2013]). These object classes and their definitions are expressed in the information model IMGeo (Informatiemodel Geografie, see Appendix A). IMGeo is based on the NEN3610 (the Dutch base model for Geo-information), and contains a mandatory and optional part. The mandatory definitions concern the BGT, and additional information on objects can be represented with optional IMGeo definitions.

All BGT data objets together cover the whole area of The Netherlands - no gaps are allowed in this coverage. While the dataset is updated and verified consistently on accounts of completeness and accuracy (the National mapping agency maintains the complete national dataset), it is developed by many organisations including municipalities, railway infrastructure owners, provinces and water boards.

Basisregistratie Topografie

The Basisregistratie Topografie (TOP10NL) dataset is mainly the result of digitized topographic maps, and is regularely updated using mainly aerial photographs (Kadaster [2020a]). It scale reaches between 1:5000 and 1:25000, and this dataset is aimed at visualization purposes. The whole dataset is updated yearly, but the images used for updates are most often collected in during the first months of each year. Besides a data model (see Appendix B), the specifications of the TOP10NL contain many acquisition rules - used to decide how to register the objects. These decisions are made by a group of experts.

3.2.3 NEN3610 and linked data representations

NEN3610 is the base model for Geo-information in The Netherlands. It is a national model based on ISO standards and compliant with European INSPIRE standards, and provides generic definitions of concepts that are re-used by the sector specific application models. These sector models are the actual models used to structure, among other, the geo-registries treated in this study case (see Figure 3.4).

Recently, the NEN3610 Linked Data Profile has been published Geonovum [2020] to facilitate the translation between UML and RDF datamodels. The document describes generic transformation rules for Dutch information models based on learned lessons and best-practices from the ISO, W3C and INSPIRE community - there were already ISO standards for generating geo-ontologies (ISO-19150-1;2) and INSPIRE guidelines are available as well [2]. A NEN3610 upper-ontology is also available publicly [3]. It defines the main classes and object attributes for 'GeoObjects' - a notation used for geographic objects within the sector

[2] http://inspire-eu-rdf.github.io/inspire-rdf-guidelines//

[3] https://github.com/Geonovum/NEN3610-Linkeddata/tree/gh-pages/NEN3610-LD

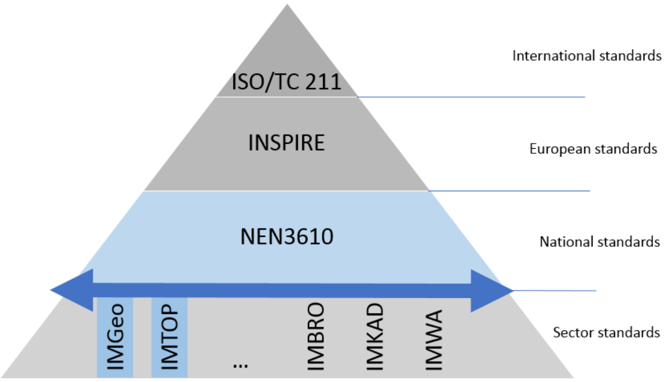


Figure 3.4: NEN3610 pyramide. Adapted from Geonovum [nd]

models. Although the classes represents a simple taxonomic structure, it can be used to connect high-level concepts from other ontologies.

The NEN3610-LD document presents three metamodels for ontology creation based on: SHACL, GWSW and COINS. Each metamodel defines its own rules for representing UML-based constructs in OWL. This highlights an important issue: there is a difference between the UML datamodel itself and the terms used to describe reality. Naturally there is a relation between both, as the terms are also used to structure classes, atributes and their values. However, both things should be kept separated according to best practices Santema and Brattinga [2018]. This vision is in line with the meaning triangle described earlier, in Chapter 2.

The idea of separating model and concepts is illustrated in the TOP10NL ontology fragment shown in Figure 3.5. Attributes from UML that were related to concept values become owl:ObjectProperty - such is the case for 'fysiek voorkomen'. Codelists become skos:collections and their values become skos:concepts. In case validation is needed, SHACL constraints can be used to link the data model (with OWL classes and properties) to the 'concept library' (with SKOS collections and concepts). Attributes that represent physical aspects (in this case, 'type landgebruik') are also modelled as OWL classes.

Thus, two different approaches towards ontology construction can be distinguished in the creation of the TOP10NL ontology: a schematic and a taxonomic approach. The schematic approach concerns the SKOS and SHACL constructs, as these are more related to the original schema of the TOP10NL (in UML). The taxonomic approach then regards the subClassOf relations that are used to model the entities (such as top10:Akkerland) that are classifications of a higher-level entity (such as top10:terrein).

3.2.4 Comparison of information models

Although IMGeo extends the CityGML model, TOP10NL as well as BGT follow the principles outlined by NEN3610. Both CityGML and NEN3610 follow ISO TC 211 modeling principles. The classes defined in the models can be traced back to the primitives established in the norm. However, a closer inspection of attributes and their values reveal many differences, as Stoter [2010] had found by comparing the BGT 0.9 with Top10NL 2.3. In recent years, however, efforts have led to a better tuning between model versions. Therefore, a brief comparison will

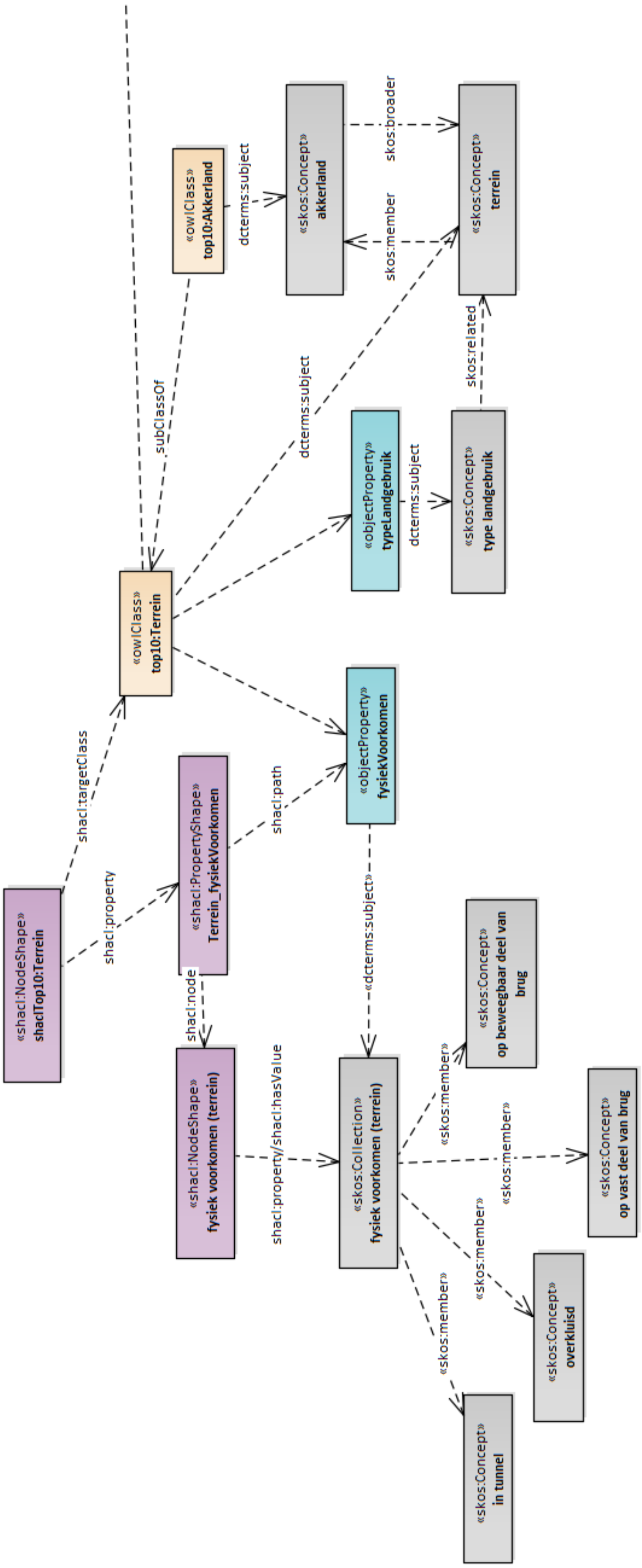


Figure 3.5: TOP10NL ontology fragment containing OWL classes and properties (in yellow and blue), SKOS concepts (in grey) and SHACL nodes (for constraints, in purple).

be made, for which the information models and documentation will be used. Data instances might also be used to highlight issues.

When attempting to align ontologies based on different datasets, one should consider the data quality of both sources. Geospatial data quality indicates whether a geospatial resource fulfills certain requirements or user needs. In other words, the data quality indicates the level of imperfection of the data and exposes how much uncertainty is involved in the process. Worboys and Clementini [2001] discuss the impact of uncertainty on the integration of geospatial data through some examples. These examples will be used to guide the comparison between the selected case study classes.

Phenomena represented by crisp concepts

Crisp concepts are those that have an clear meaning, often based on administrative boundaries - countries, cities, neighborhoods -, or official functions related to an object. Such is the case for objects within the classes 'Functioneel Gebied' and 'Registratief Gebied' from the TOP10NL and BGT. In both datasets, the geometric representation of these objects is not necessarily related to that of objects from different classes. Therefore, these objects can be layered to accommodate different views.

Vagueness due to incomplete representation

In Worboys and Clementini [2001] incomplete feature representation refers to missing geometries, often due to data conversion. Given the official nature of the geo-registries and the quality standards they adhere to, this subject is not considered relevant for this case. Nonetheless, incomplete representation can also apply to attribute values. In both registries terms related to functional and physical aspects of the object are given in codelists (or skos:collections in the linked data version of the TOP10). Usually this includes the value 'overig', indicating the term needed is not included in the official vocabulary. Although this restricts the semantics, most codelists provide enough attribute values to allow comparisons between the semantics of objects. However, within the BGT certain degrees of freedom are allowed through the 'plus-type' attribute values - which are optional. While the 'plus-type' is supposed to convey more specific information regarding an object, the values for this attribute sometimes overlap with those of TOP10NL objects (at least syntactically, see Figure 3.6). This means information could go 'lost', because it is not mandatory to specify it.

Vagueness due to dynamic aspects of phenomena

All spatial phenomena recorded in datasets have some type of variability related to them: they might undergo changes over time. With man-made object types (such as functional/geographic regions, buildings or furniture, for example) it is possible to record object mutations that indicate whether something has stopped existing in the observed world or has changed in some way. However, there is a second type of variability that cannot be solved

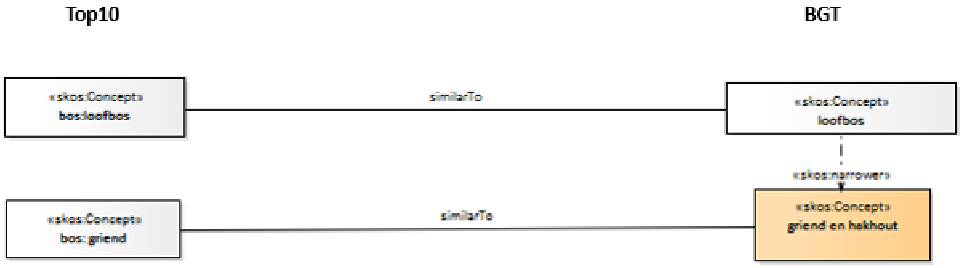


Figure 3.6: Example of possibly problematic mapping, with an optional IMGeo property value in orange.

in such a way. It relates to objects with fuzzy boundaries (lakes and rivers, among others) that constantly change depending on external conditions. In the Top10NL this fuzziness is not accounted for: with lakes, only their minimal areas (which are homogeneously covered in water for an sufficient amount of time) are registered. The methods used to decide this are not explicitly mentioned in the documentation. The BGT, besides also registering these minimal areas, has a special class ’OndersteunendWaterdeel’ with geometries that represent the ’time-varying’ region of a waterbody.

Inherently vague phenomena and conflicting representations

Sometimes the spatial objects represented in the dataset are inherently vague, because the features they represent cannot be captured without involving some sort of interpretation. This is usually the case with non man-made geographic objects. The BGT and TOP10NL Terrain classes contain many such examples. For example, both datasets contain objects of type ’loofbos’ and ’naaldbos’, but these human-defined terms based on specific conventions or rules followed by both datasets. These rules (inwinningsregels) are described in the documentation. As an example: in the TOP10 product specifications Kadaster [2020b], a terrain is typed as ’loofbos’ if it is least 90% covered with deciduous trees and has a minimum area of 1000 m2. In the specifications of the Ministerie van Infrastructuur en Milieu [2013] the definition is even more open to interpretation: a terrain is typed as ’loofbos’ if the amount of deciduous trees is high enough to enclose the area.

Discrepancies and vagueness in the definitions of these object classes are not the only problem. Objects defined as forests are actually collections of trees which are aggregated according to certain metrics - such as the density and distribution of trees across some spatial ’unit’. This unit might be tied to administrative boundaries or natural boundaries (that are only perceived at the resolution of the dataset). Hence, the semantic classes defined in the datasets might only be relevant for a specific resolution. Therefore, even if both datasets use the same terms to describe physical objects (such as terrains of type ’heide’), the boundaries of these objects might not always coincide. For example, in Figure 3.7 the TOP10NL defines a portion of land as one object, while in the BGT the division seems based on smaller physical patterns (probably ditches) in the land.

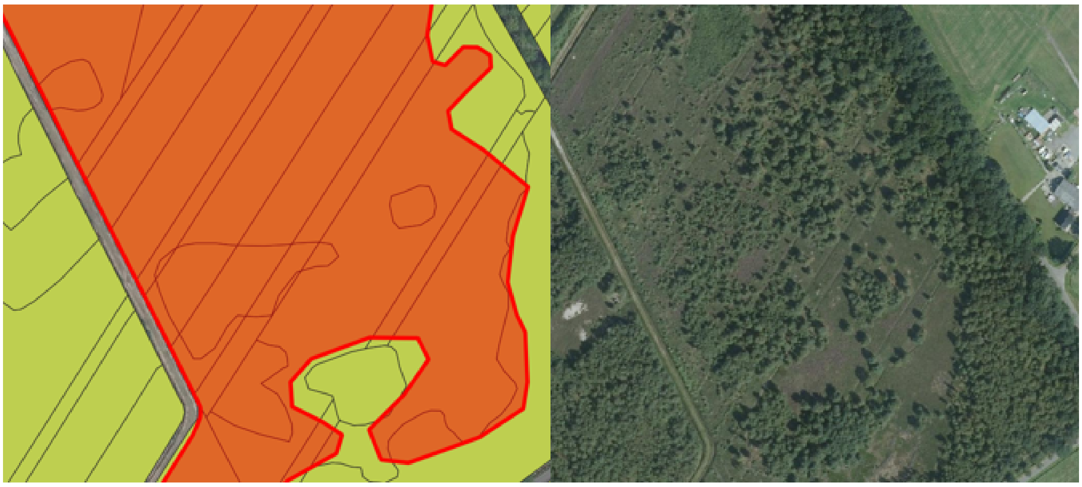


Figure 3.7: On the left: overlay of top10 object (red) with BGT objects (green); On the right: satellite view of the same area, in the same year as the object registration

And even if both datasets have been constructed from finely grained resolution, their scales might still represent course-grained objects depending on the semantics that they support. This is the case with some BGT Terrain objects: although the dataset has a larger scale than the Top10NL, it seems to provide less detailed terrain information in rural areas as compared to urban areas (see en Landschap [2019]).

Another aspect to consider is the geometric feature types used. It would be difficult to integrate geospatial data on an instance level if there is no spatial relationship of any kind between the instances of two classes - or, if defining such a relation would lead to a lot of ambiguity. Such could be the case with the 'Waterdeel' class. In the BGT, all features from this class are represented as polygons as well as lines. In the top10, however, certain feature types are represented as polygons (watervlakte), others as lines (greppel, droge sloot), and some might be represented in either one of these ways (waterloop). Even more, many of the line representations in the TOP10 dataset do not seem to follow any consistent division. Many 'loose' segments are attached to one another. Thus, it is not possible to directly compare these features to those in the BGT based on their geometries (see Figure 3.8).

3.2.5 Suitability of matching strategies

The suitability of matching strategies is mainly influenced by the type of correspondences that are needed for the application at hand. In this case, data interlinking is the main objective. Matching is used as means to support better interlinking.

After reviewing both models, the main conclusions can be summarized by the following points:

- Shared vocabulary and similar structures: through the NEN3610 generic class definitions, both models are already linked on a higher level. Moreover, their structure is very similar: each class has attributes describing physical and functional aspects with predefined values, represented in a hierarchical structure - the full expressivity of OWL

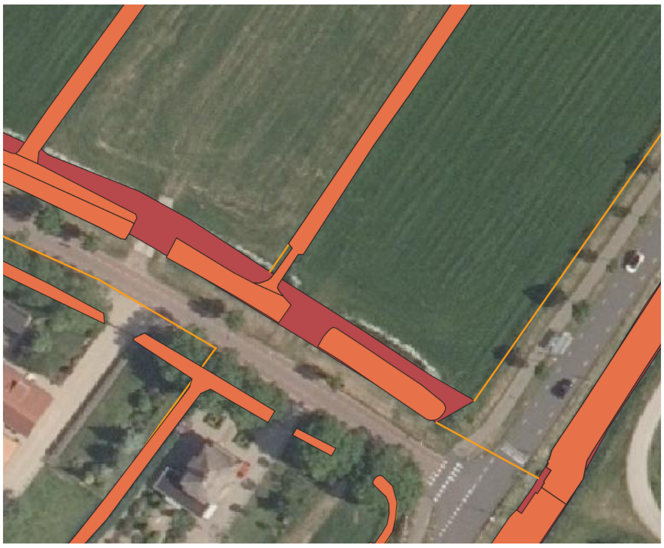


Figure 3.8: Example of differences between BGT Water objects (polygons) and TOP10NL Water objects (lines)

is not used. While some classes might have more specific attributes, these usually do not refer to their semantics directly.

- Data models with many (syntactically) overlapping terms: as they handle similar domains, many concepts related to the physical and functional aspects of objects seem to overlap - mostly within related classes such as Water-Waterdeel and Terrein-BegroeidTerrein/OnbegroeidTerrein. However, acquisition rules suggest very different classification methods. Understanding the extent to which these terms overlap could provide more insights in their meaning - instances should be used to this end.

- Large number of instances, different representations: the datasets contain many information objects based on the models. However, different scales, representations (boundaries and geometric types) and lack of valid time information on the real-world situation might hinder a complete comparison. Regardless, comparing instances could help indicate if similar terms are also used that way.

Thus, matching strategies should be able to compare the similarity between concepts described in a (mostly taxonomic) structure supported by a very large number of instance data which can only partially be used (due to differences in representation - on geometric as well as semantic level). In the next section a few such methods will be discussed in more detail.

Naturally, some characteristics of the selected datasets might be exclusive to their situation - in which case the methods chosen might not apply to other registries in the same way. However, it has been observed that the relation between the case study datasets and the ISO standards on geographic information is well founded through the NEN3610 standard. Therefore, the assumption is made that the suitable matching strategies selected for the case also apply (at least partially) to other similar geo-registries.

3.3 Matchers

This section presents some of the most common methods used for often-used matching strategies, based on literature findings.

3.3.1 Terminology: String-based and language-based methods

Terminological matchers are usually employed to improve other matchers. Name-based techniques often focus on string comparison, regarding words or sentences as sequences of letters rather than concepts. The most common methods of string comparison are the substring test and the edit distance. As the name suggests, a substring test involves finding out whether a sequence of letters representing a concept can be located within the string representing another concept. If so, both concepts are similar. The similarity value can be calculated based on the ratio of matching characters between the concepts. Edit distance uses a completely different approach: the similarity between concepts is defined by the minimal set of operations (insertions, substitutions and deletions) necessary to transform one string in the other. Methods based on edit distance are especially useful in cases of spelling mistakes – these mistakes will not weight much as is the case with substring matching. Many variations of this method exist - including Levenshtein Distance, Hamming distance and Jaro-Winkler distance -, each with its own set of allowed character operations. Although string-based methods are well-suited for syntactic comparisons they cannot identify homonyms (one word with multiple meanings) and synonyms (different words with the same meaning).

Language-based methods can solve the issue of synonyms by comparing the concept words to entries in lexicons or thesauri. A well-known example is WordNet, a lexical database containing words grouped in sets of synonyms (called synsets). However, even if two different words are found to be synonyms, the meaning (or sense) behind their use might still be different. To assess this using language-based methods requires more contextual information – such as textual descriptions or comments related to the concepts in question. Natural-language processing (NLP) could then be used to extract terms from these descriptions, and the similarity of all terms related to two concepts could be used to determine whether or not they are synonyms/homonyms and how similar they are. Alternatively, additional matchers based on the structure of the ontology could be used to deduce this information.

3.3.2 Structural: Graph-based and taxonomic methods

As discussed in Chapter 2, graph-based techniques take ontologies as input and transform them into labelled graphs: objects and subjects become nodes and predicates become edges of the graph. The similarity between entities is dependent on their position in the graph, with the underlying assumption that neighbouring nodes are similar in some way. The graph-based methods are further subdivided based on the type of relations they can handle. Most methods make use of taxonomic relations (by creating graphs based on rdfs:subClassOf relations within ontologies), and some also make use of mereological information (by incorporating part/whole relations, usually expressed through custom predicates). Additionally, it is also possible to analyze the similarity between all relations expressed in the ontologies and use this information to determine the similarity between the classes they relate to. Regardless of the type of relations considered, graph-based methods are usually grouped in node-based and edge-based approaches.

Edge-based approaches calculate the distance between two concepts by measuring the path length between the nodes representing these concepts. Rada et al. [1989] proposed that the shortest path linking both nodes to a common ancestor should be used to indicate the distance, which in turn determines the semantic similarity. In an unweighted model, all edges contribute equally to the final distance. However, some aspects of the ontologies could indicate that weights are appropriate. Firstly, if a node is nested deeply in the hierarchy, it might convey more specific knowledge– hence, the distance to its parents will be shorter.Wu and Palmer [1994] accounted for this by incorporating the distance between the common ancestor of the nodes and the root node of the ontology into their similarity calculation. Secondly, if multiple relation types - such as subclass and part/whole - are represented, weights should also be considered. Thirdly, nodes that are part of a very dense subgraph might be more similar to one another, as was observed with the local density effect Richardson and Smeaton [1995].

Node-based approaches do not suffer from these issues, as they rely less on the structure of the graph and more on the information that is shared between concepts and their ancestors or descendants (such as properties). One of the best-known node-based approaches is that of information content [Resnik, 1999]. In this approach, edges are only used to represent connections between nodes – with no weights or meaning attached to it. Each node's concept has an 'information content' (IC) attached to it. This is inversely related to the probability of encountering the concept in a text corpus. In other words: concepts whose node is high in the graph hierarchy have a lower IC – they are more general and therefore, less informative. The similarity between two nodes is then defined as the extent to which they share information, or the IC of their most informative common ancestor (MICA). The main limitation of this approach is the nodes sharing the same lowest common parent will also share similarity values. For domain ontologies linked through a custom vocabulary, this would mean most concepts share the same similarity value – which is of little use in practice. Jiang and Conrath [1997] attempted to counter this effect including the IC of the concepts being compared in their calculations. However, these techniques were developed and tested for WordNet (a lexical database) and as such might not reflect other particularities related to geospatial definitions – such as asymmetric relations (concept 'A' is more similar to 'B' than 'B' is to 'A').

$$\mathrm{Dist}(w_1, w_2) = \mathrm{IC}(c_1) + \mathrm{IC}(c_2) - 2 \cdot \mathrm{IC}(\mathrm{LCA}(c_1, c_2)) \tag{3.1}$$

With this in mind, Rodríguez and Egenhofer [2004] developed the Matching-Distance Similarity Measure. This consisted of a feature-matching technique, based on the idea that every concept in an ontology can be describe in terms of their features. Similarity between two concepts is calculated based on the intersection and difference of the collection of features attributed to them. The researchers distinguished between different types of features (parts, functions and attributes). Each feature type was linked to a similarity function, based on the ratio model described by Tversky (1977). Their solution differed from Tversky's by quantifying the relative importance of different features between two concepts by using the distance between the concepts in a hierarchy in the calculations. This would account for the asymmetric nature of taxonomic relations, such that it can be that: S(c1, c2) != S(c2, c1). However, for the MDSM to produce good results, the ontology must be structured in such a way that its concepts have enough distinctive features describing them.

$$S(c_1, c_2) = \omega_p \cdot S_p(c_1, c_2) + \omega_f \cdot S_f(c_1, c_2) + \omega_a \cdot S_a(c_1, c_2) \tag{3.2}$$

$$S_t(c_1, c_2) = \frac{|C_1 \cap C_2|}{|C_1 \cap C_2| + \alpha(c_1, c_2) \cdot |C_1 \setminus C_2| + (1 - \alpha(c_1, c_2)) \cdot |C_2 \setminus C_1|} \tag{3.3}$$

In cases where the structure of the ontology most likely does not support such measures, graph-based techniques can still be used on top of other matchers. Sunna and Cruz [2007] created an alignment method for geospatial ontologies through which base similarity calculations could be enhanced using two node-based techniques: The Descendant's Similarity Inheritance (DSI) or the Sibling's Similarity Contribution (SSC) method. With DSI, the similarity between two concepts is determined by their base similarity and the weighted sum of the base similarities of their parents – going up until the root node of each ontology. The larger the path length between the parents and the concepts, the less their similarity counts: the similarity between 'grandparent' concepts counts less than the similarity of 'parent' concepts. The formula is given below. In the SSC method, base similarities are similarly enhanced by comparing the siblings (nodes of the same parent) of one concept to those of the other and adding the maximum similarity value of this comparison to the similarity calculation of the two concepts. (formula below). The main limitation of both DSI and SSC methods is that they assume parents and siblings of concepts also share similar traits – if this is not the case, then the methods do not contribute much to the base similarity.

$$\begin{aligned} \mathrm{Sim}(C, C') = {} & \mathrm{MCP} \cdot \mathrm{base\_sim}(C, C') \\ & + \tfrac{2(1-\mathrm{MCP})}{n(n+1)} \textstyle\sum_{i=1}^{n} (n+1-i) \cdot \mathrm{base\_sim}(\mathrm{parent}_i(C), \mathrm{parent}_i(C')) \\ \text{where} \quad & n = \min(\mathrm{path\_len\_root}(C), \mathrm{path\_len\_root}(C')) \end{aligned} \tag{3.4}$$

$$\begin{aligned} \mathrm{Sim}(C, C') = {} & \mathrm{MCP} \cdot \mathrm{base\_sim}(C, C') \\ & + \tfrac{1-\mathrm{MCP}}{n} \textstyle\sum_{i=1}^{n} \max(\mathrm{base\_sim}(S_i, S'_1), \cdots, \mathrm{base\_sim}(S_i, S'_m)) \\ \text{where} \quad & n = \mathrm{sibling\_count}(C) \\ & m = \mathrm{sibling\_count}(C') \end{aligned} \tag{3.5}$$

3.3.3 Extensional: Instance-based methods

Before introducing extensional matching, the importance of such methods (within the context of geospatial applications) should be reflected upon. To this end, the concepts of semantic similarity and semantic relatedness will be discussed.

Semantically similar concepts are related by synonymy or hypernymy/hyponymy - in other words, they share a 'is-a' or 'type-of' relation. Relatedness, on the other hand, conveys more general associations between objects that are not necessarily similar. As an example: the concept *house* might be similar to *apartment* in terms of affordance (both are residential units), and related to the concept *garden*. Such relatedness could already have been stated in the data model or ontology (the garden could be 'part of' the house), but it could also be detected by observing the topological relations between data instances. And this is precisely why relatedness is important in geospatial applications: it allows understanding the different logical layers used to represent the physical world (as described in Section 2.1). And by understanding these layers and the extent to which they can be fused, more information could be retrieved on the different facets of a physical space - leading to less duplication. The importance of relatedness in geospatial semantics has only recently begun to attract more attention (Kokla and Guilbert [2020]; Ballatore et al. [2014]; Hecht et al. [2012]).

In the case of integrating two data models, such information could be used in conjunction with terminological matching to distinguish between homonyms, for example. The study cases already provide some measures of relatedness - by linking basic object types to NEN3610 classes. While graph-based matching can take advantage of these links during alignment, it does not allow to explore the relatedness between more specific subclasses or property values.

Extensional matching can help shed more light on the relation between these specific subclasses based on the instances that they share across ontologies. This technique was adopted by [Parundekar et al., 2010] for matching GeoNames classes with DBPedia classes. The authors use preexisting owl:sameAs links between GeoNames and DBPedia instances to create instance pairs. Only the instances that are part of instance pairs are considered in this framework. Each instance pair contains the characteristic attributes/values for both datasets (which they call a restriction class). An example of an instance pair is: [ont1:School, ont2:EducationalInstitution]. Each instance pair represents a potential match between classes/attribute values of the datasets, and are referred to as a hypothesis. Then, for each hypothesis all instances belonging to the restriction class from ontology 1 (r1) and ontology 2 (r2) are counted. The number of instance pairs that fulfill the hypothesis (belonging to both r1 and r2) is also counted. The idea is illustrated in Figure 3.9. Divided by the isolated restriction class counts this leads to two measures P and R which are used to determine the set-containment relations (whether the concepts from the hypothesis are equivalent, subsumed by one another or disjoint). As far as is known this is one of the only attempts in the geospatial domain to align ontologies based on existing links between shared instances.

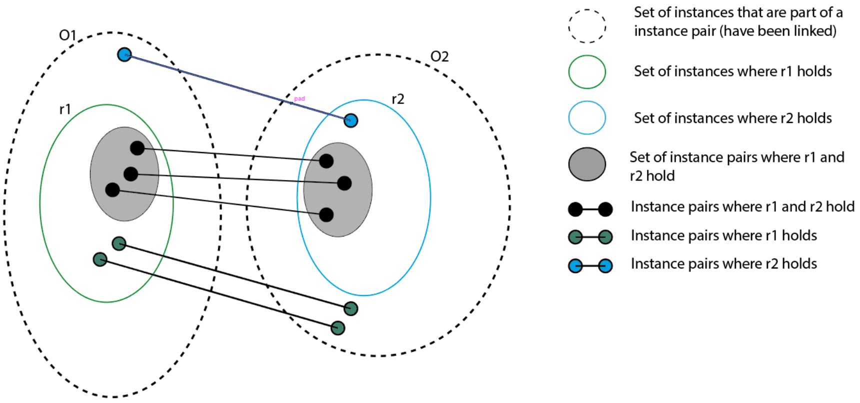

O1
O2
r1
r2
Set of instances that are part of a instance pair (have been linked)
Set of instances where r1 holds
Set of instances where r2 holds
Set of instance pairs where r1 and r2 hold
Instance pairs where r1 and r2 hold
Instance pairs where r1 holds
Instance pairs where r2 holds

Figure 3.9: Example of hypothesis testing for aligning two ontology classes based on instance pairs. Adapted from Parundekar et al. [2010]

4 Conceptual architecture

In Chapter 2 an overview was given of the main aspects involved in semantic integration frameworks. Many matching techniques can be used to generate alignments, and choices will depend on the application at hand. In Chapter 3 the overview is given of some matching strategies which are relevant to the case study. This chapter explains in more detail how matching is performed, the expected results and the types of mapping language formats that could be useful for expressing the alignments. Afterwards, the query framework will be introduced. This framework will be used to evaluate the usefulness of the alignments. It should be remembered data interlinking and ontology matching are two processes that can take advantage of each other recursively: ontology matching can use extensional knowledge (in this case, topological links between geometries), and data interlinking can use the results from ontology matching (the alignments) to improve itself.

4.1 Alignment framework

In this research geospatial data integration will be explored by combining matching methods and expressing these results semantically, through user-defined relations - in an attempt to overcome the limitations of the the 'owl:equivalentClass' and 'owl:sameAs' constructs on spatial objects. This involves defining a framework for the alignment - based on the conclusions of Chapter 3. The basic outline of the framework is shown in Figure 4.1 and will be discussed below.

4.1.1 Structural and string-based matching

String-based matching will allow identifying similar words. For this, different edit distance formulas could be used as explained in Section 3.3. Jaro-winkler has been selected as it is more appropriate for smaller strings (Cohen et al. [2003]), and because it has a normalized score - where 0 means there is no similarity and 1 that the words are identical. There are many implementations of edit distance algorithms available, and in this case the Jellyfish [1] library for python will be used. The words used will be the rdfs:label values from the owl:Class entities (or the skos:prefLabel of skos:Concepts, in the case of the TOP10NL attributes).

The string-based matching will then serve as input for the structural matching. A variation of the formula from Equation 3.4 will be used, as it considers the similarity of parent nodes to impact the final score. This is important as it gives more weight to the relatedness between object classes. Similar concept words in the BGT and TOP10NL subsumed by different

[1] https://pypi.org/project/jellyfish/

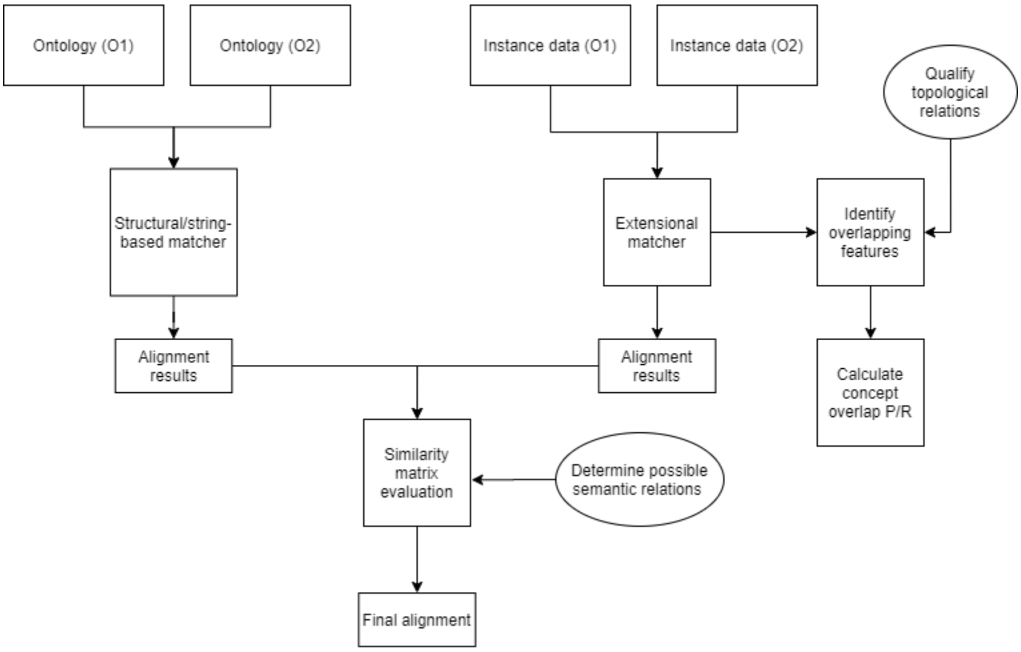

Ontology (O1)
Ontology (O2)
Instance data (O1)
Instance data (O2)
Qualify topological relations
Structural/string-based matcher
Extensional matcher
Identify overlapping features
Alignment results
Alignment results
Calculate concept overlap P/R
Similarity matrix evaluation
Determine possible semantic relations
Final alignment

Figure 4.1: Alignment framework outline

NEN3610 classes are supposed to be less similar than concept words subsumed to the same NEN3610 class. The parent nodes of each TOP10NL and BGT class are retrieved by creating a subgraph of both ontologies using RDFLib [2], containing only triples with the rdfs:subClassOf predicate. This subgraph is transformed into a directed graph using NetworkX [3], to get the path length between all nodes. However, not all concepts that are relevant for the comparison are modeled using rdfs:subClassOf. The TOP10NL ontology contains skos:Concepts related to other attributes through SHACL constraints, and some of these concepts might be just as relevant as the ones expressed using rdfs:subClassOf relations. An example is the skos:Concept ’haven’ related to the attribute top10:functie within the domain Waterdeel- in the BGT ’haven’ is expressed as a rdfs:subClassOf of watervlakte and Waterdeel. This exposes the issue already mentioned in Chapter 3: functional and physical aspects of objects cannot be clearly separated. Because of such cases, skos:Concepts from the TOP10NL ontology should also be considered during structural matching. Their parent nodes will correspond to the sh:targetClass of the main NodeShape they connect to (see Figure 3.5 again for the relation between SHACL shapes and the OWL classes of the TOP10NL ontology).

It should be noted that the string-based and structural matching described above are used to highlight and discuss the contributions and limitations of such approaches when dealing with geospatial entities. The results from this process are not necessarily ’right’, as no evaluation is carried out on them based on ’ground truth’ alignments - as is done during the OAEI campaigns. However, the output will be manually inspected and - if necessary - adjusted to provide more appropriate similarity measures to be combined with the results of the extensional matching.

[2] https://rdflib.readthedocs.io/en/stable/

[3] https://networkx.github.io/

4.1.2 Extensional matching

Extensional matching is actually a necessary matching step, as analysis given in Chapter 3 reveals similarity between property values does not guarantee all features sharing those values to overlap consistently. The extensional matcher first needs to determine possible relations between the instances of both datasets. This step involves qualifying topological relations. As discussed in Chapter 3, different data granularities and cognitive world views can lead to certain instances being aggregated in the other dataset. Thus, four basic relations are identified:

- 1-to-1 relation between top10 and bgt instance
- 1-to-many relation between a top10 instance and many bgt instances
- 1-to-many relation between a bgt instance and many top10 instances
- many-to-many relation between top10 and bgt instances

The many-to-many relations would require finding patterns that could justify typification of some sort - how this would be done and how it would be implemented is out of scope for this research. Thus, only 1-to-1 and 1-to-many relations will be analyzed. Additionally, the 1-to-1 relation could be separated into 'equals', 'contains' and 'contained'. In both cases, the overlap between instances of the BGT and Top10NL is evaluated according to the code in X. If the overlap between instances is higher than the heuristically defined threshold, the relation between them will hold (this is illustrated in 4.2). The threshold is necessary due to different accuracy levels between the datasets - two 'equal' instances in the BGT and TOP10NL will rarely be truly geometrically equal, but they will overlap considerably.

The extensional matching used here extends the approach described in Chapter 3. Instead of using instance pairs linked through owl:sameAs predicates, the relations described in the previous paragraph are used. The main difference is that the probability measures P and R (which are used to determine the set containment of concepts) can be calculated for the instance pairs linked through a specific topological relation ($ProbT$ and $ProbB$) as well as for the instance pairs linked through any relation ($ProbT - All$ and $ProbB - All$). Instance pairs will usually be composed of one TOP10NL attribute value (from 'type' attributes) and one BGT attribute value (from either 'bgt-type' or 'plus-type' attributes). Below, these attributes are simply referred to as 'subclass', as they are modeled as owl:Class. While the TOP10NL has other attribute values which could be useful (such as Waterdeel 'functie'), most of these secondary attributes are optional, and often go unfilled - this is the case with 'gebruiksdoel', which is very similar to the attribute found in he BAG. Unfortunately, it is rarely filled so it is not worth considering. The same can be said for the TOP10NL fysiek voorkomen attribute of Terrein (there are only 4, related to another class kept out of scope here - bridges); and fysiek voorkomen of waterdeel will also be ignored - because it seems these are only used for linestring geometries, which are not looked at. Basically, only the values that characterize the objects (their types) can reliably be used - and these often represent defining features.

During matching, count is kept on the instance pairs and their relations (equals, contains, is contained, aggregates, is aggregated). Then, for every instance pair in the TOP10NL and BGT, the conditional probabilities are calculated as follows (example concerning only 'equals' relations):

$$P(Top10_{classX}|_{equals}BGT_{classY}) = \frac{P(Top10_{classX} \cap BGT_{classY})}{P(BGT_{classY})}$$

$$P(BGT_{classY}|_{equals}Top10_{classX}) = \frac{P(Top10_{classX} \cap BGT_{classY})}{P(Top10_{classX})}$$

Where:

- $P(Top10_{classX}|_{equals}BGT_{classY})$ - or $ProbT$ is the probability that a top10 object of class X has an 'equals' relation with a BGT object of class Y.
- $P(BGT_{classY}|_{equals}Top10_{classX})$ - or $ProbB$is the probability that a BGT object of class Y has an 'equals' relation with a Top10 object of class X.
- $P(Top10_{classX} \cap BGT_{classY})$ is the set of objects with relation 'equals', where the top10 class is X and the BGT class is Y.
- $P(BGT_{classY})$ is the set of BGT objects of class Y that have 'equals' relation with top10 objects.
- $P(Top10_{classX})$ is the set of Top10 objects of class X that have 'equals' relation with BGT objects.

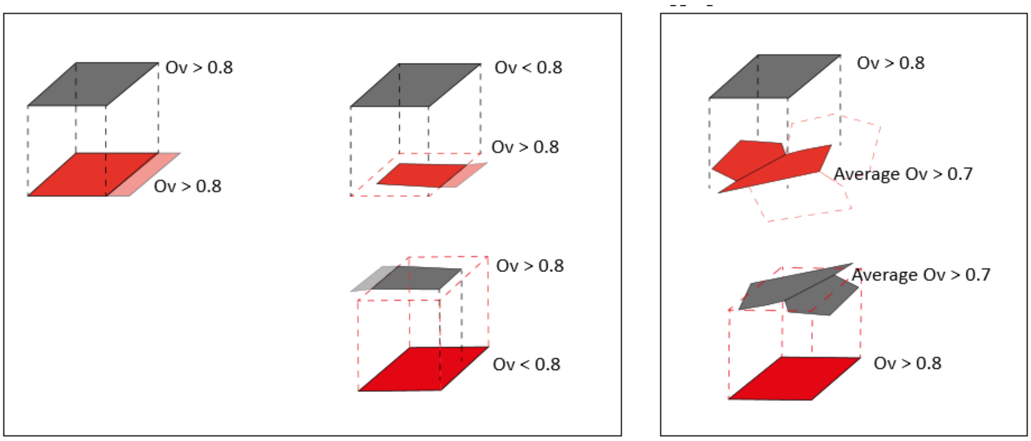
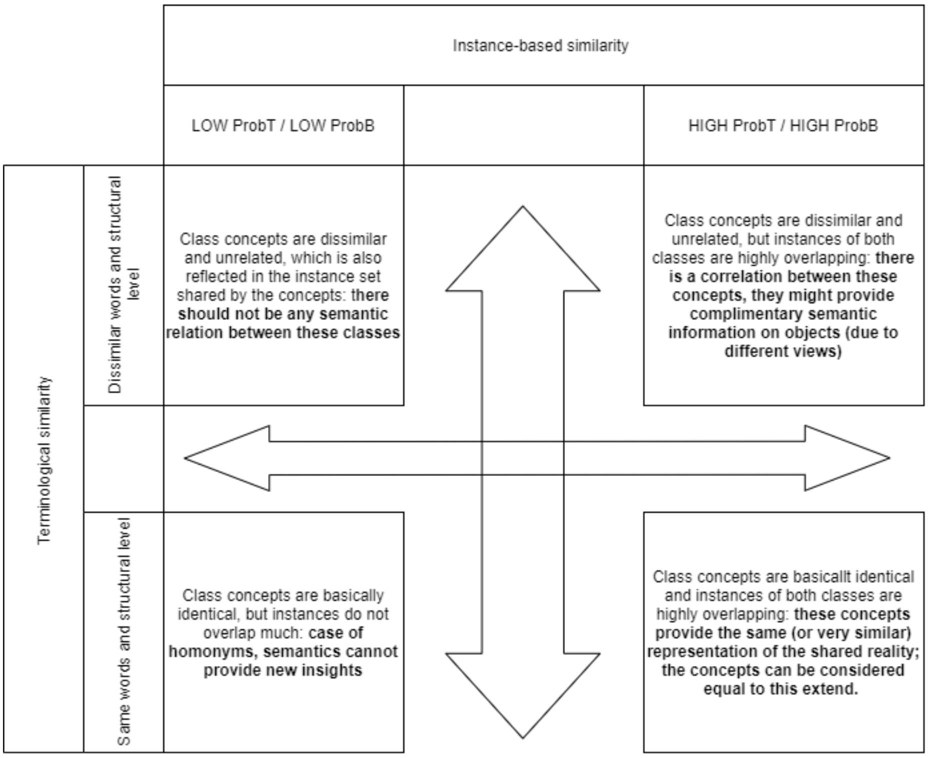


Figure 4.2: Possible relations between BGT and Top10NL instance data

These two measures ($ProbT$ and $ProbB$) allow saying something about the relation between ontology concepts based on the instances they share through a topological link. For example, if both probabilities are above a certain threshold it can be said the two concepts are 'equal', when their instances are topologically linked based on the same granularity. It is also possible to disregard specific topological links by determining $ProbT$ and $ProbB$ based on any shared topological link. Then, if these $ProbT - All$ and $ProbB - All$ are above the threshold, one could say the two concepts are 'equal' when their instances are topologically linked in any way.

On itself, this does not provide much insight. However, by combining this information with that of the previous matchers it is possible to tell if the concepts portray a similar


Instance-based similarity
LOW ProbT / LOW ProbB
HIGH ProbT / HIGH ProbB
Terminological similarity
Dissimilar words and structural level
Same words and structural level

Class concepts are dissimilar and unrelated, which is also reflected in the instance set shared by the concepts: **there should not be any semantic relation between these classes**

Class concepts are dissimilar and unrelated, but instances of both classes are highly overlapping: **there is a correlation between these concepts, they might provide complimentary semantic information on objects (due to different views)**

Class concepts are basically identical, but instances do not overlap much: **case of homonyms, semantics cannot provide new insights**

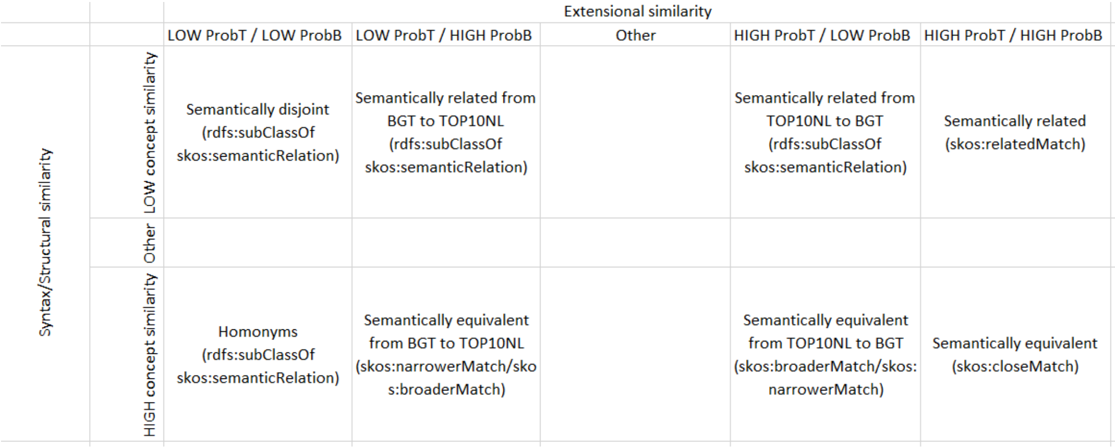
Class concepts are basicallt identical and instances of both classes are highly overlapping: **these concepts provide the same (or very similar) representation of the shared reality; the concepts can be considered equal to this extend.**

Figure 4.3: Possible semantic relations between BGT and Top10NL concepts

representation of the world (the concepts have similar syntax and are related within the data models) or if there is correlation between the concepts (when they are not similar or related on the schema-level, but they do share a set of instances). The second scenario could indicate 'different views' on the same reality, exposing associative relations that could not be captured within the data model. The semantic relations that can be established based on these two matchers are portrayed in 4.3. However, the similarity values are on a gradient, so whether a semantic relation between two classes is made will depend on the threshold that is set.

Regarding temporal attributes: both BGT and TOP10NL do not contain information on the material history of objects (it is only specified when objects were registered in the system). The TOP10, however, does contain a 'bronactualiteit' attribute, specifying the creation date of the data sources used (aerial images, vector data, etc). This date is useful as it affects the shapes of the geometries in the dataset - which are used in the instance-based matching. It would be possible to then select only BGT features from around this date (if we assume the object time is not too different from the real-observed time). However, many TOP10NL objects with geometries based on older sources have undergone mutations (noted in the attribute 'mutatietype'). Therefore, their functions might not reflect those of the 'bronactualiteit' date on which their geometry is based. This is a limitation of the method and cannot be handled in this phase.

For the resulting alignments the sum of all relations is used (equals, contains, aggregates, etc) to determine the $ProbT \quad ProbB$ ratio and whether two concepts are related. This is done because otherwise results are scewed, for example: some very broad physical material like 'asfalt' being subsumed by a specific function 'Tennispark' just because they often match one to one.



		Extensional similarity				
		LOW ProbT / LOW ProbB	LOW ProbT / HIGH ProbB	Other	HIGH ProbT / LOW ProbB	HIGH ProbT / HIGH ProbB
Syntax/Structural similarity	LOW concept similarity	Semantically disjoint (rdfs:subClassOf skos:semanticRelation)	Semantically related from BGT to TOP10NL (rdfs:subClassOf skos:semanticRelation)		Semantically related from TOP10NL to BGT (rdfs:subClassOf skos:semanticRelation)	Semantically related (skos:relatedMatch)
	Other					
	HIGH concept similarity	Homonyms (rdfs:subClassOf skos:semanticRelation)	Semantically equivalent from BGT to TOP10NL (skos:narrowerMatch/skos:broaderMatch)		Semantically equivalent from TOP10NL to BGT (skos:broaderMatch/skos:narrowerMatch)	Semantically equivalent (skos:closeMatch)

Figure 4.4: Possible semantic relations between BGT and Top10NL concepts

4.1.3 Mapping language

The final alignment should contain, for each pair of concepts from the BGT and TOP10NL classes, a mapping specifying the semantic relation found between them. Custom relation properties can be created to this end, as illustrated in Figure 4.4. Some can be based on existing properties from the SKOS vocabulary, others can be created using extra OWL assertions (see Listing 4.3). The different relations that were identified can be described as follows:

- Semantically disjoint: Relation between two concepts that are not in any way similar, and do not share a set of overlapping instances.
- Homonyms: Relation between two very similar concepts, that have no shared set of instances - while the words might be similar, the actual concepts refer to different things.
- Semantically related: Relation between two concepts that are not similar, but do share a set of instances - this indicates an associative relation, exposing more information on the involved objects
- Semantically equivalent: Relation between two concept that are very similar, and share a set of instances
- Semantically related from X to Y: Relation between two concept that are not similar, but where the bigger part of the set of instances of X overlap with those of Y (the inverse is not true).
- Semantically equivalent from X to Y: Relation between two concept that are very similar, and where the bigger part of the set of instances of X overlap with those of Y (the inverse is not true).

Due to the use of different thresholds for all matchers and the inherent vagueness involved in extensional matching, no mapping between two entities of the ontologies is guaranteed to be correct. Some form of confidence level should be attached to each correspondence.

Hence, direct OWL or SKOS mappings will not suffice. On the other hand, the correspondences are fairly simple as they most likely involve basic instance pairs with concepts that represent object 'types' - other defining attributes are often optional and less recurrent, as mentioned before. The Alignment format is suited for such situations: it can create simple correspondences between two entities with a confidence value attached, or more complex correspondences using EDOAL (between two scoped entities with specific attribute values, for example). Both examples are shown below, in 4.1 and 4.2. Each 'align:measure' will be a function of ($ProbT - All$ and $ProbB - All$)/($ProbT$ and $ProbB$) - were ($ProbT - All$ and $ProbB - All$) are the probabilities for all matches joined (from aggregations, contains, equals, etc).

Listing 4.1: simple Alignment format mapping example: TOP10NL Bungalowpark objects are most likely equal to BGT Bungalowpark objects

```
<map>
  <Cell>
    <entity1 rdf:resource='http://brt.reg.nl/def/top10nl#Bungalowpark'/>
    <entity2 rdf:resource='https://geostandaarden.nl/imgeo#Bungalowpark'/>
    <relation>fr.inrialpes.exmo.align.impl.rel.EquivRelation</relation>
    <measure rdf:datatype='http://XMLSchema#float'>0.85</measure>
  </Cell>
</map>
```

Listing 4.2: EDOAL mapping example: TOP10NL MeerPlas objects with 'function' value Haven are equal to BGT Haven objects

```
<align:Cell rdf:about="#cell1">
  <align:entity1>
    <Class>
    <Class rdf:about="http://brt.reg.nl/def/top10nl#MeerPlas" />
    <AttributeValueRestriction>
      <onAttribute>
        <Relation rdf:about="http://brt.reg.nl/def/top10nl#functie" />
      </onAttribute>
        <comparator rdf:resource="&edoal;equals" />
      <value>
      <Instance rdf:about="http://brt.registraties.nl/id/begrip/Haven" />
      </value>
    </AttributeValueRestriction>
    </Class>
  </align:entity1>
  <align:entity2>
  <Class rdf:about="https://geostandaarden.nl/imgeo#Haven_Waterdeel" />
  </align:entity2>
  <align:relation>=</align:relation>
</align:Cell>
```

Listing 4.3: Show the custom relations expressed as OWL/SKOS sub-properties (for the relations: semantically related from BGT to TOP10NL/TOP10NL to BGT)

```
rel:relatedTo rdfs:subClassOf skos:semanticRelation ;
```

```
    a owl:AsymmetricObjectProperty ;
    rdfs:label "The property that determines that two concepts are
    not semantically similar, but that the subject is related
    to the object through an associative relation of some sort
    (the inverse is not true). " ;
    rdfs:domain skos:Concept ;
    rdfs:range skos:Concept .
```

4.2 Query framework

The query framework will involve retrieving linked data already available through federated queries based on geospatial links and processing this according to the same rules defined for the extensional matching (based on geospatial overlap). The alignments could be used to check for consistency or to retrieve more information - depending on the results of the matching process. Procedural languages such as SPARQL constructs and SPIN/SHACL are more appropriate for queries, as federated queries are required. SPARQL Construct queries, specifically, can be used within other languages - such as SPIN and SHACL, so that transformations or other operations can be handled dynamically during query time.

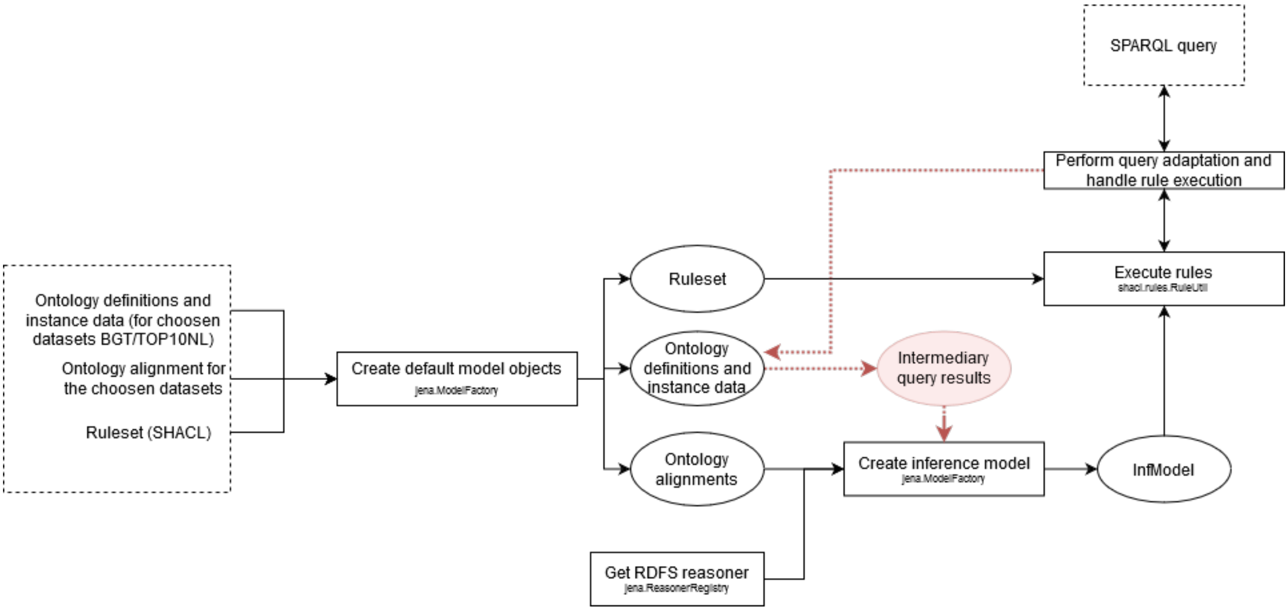

SPARQL query
Perform query adaptation and handle rule execution
Ontology definitions and instance data (for choosen datasets BGT/TOP10NL)
Ontology alignment for the choosen datasets
Ruleset (SHACL)
Create default model objects
jena.ModelFactory
Ruleset
Ontology definitions and instance data
Ontology alignments
Intermediary query results
Execute rules
shacl.rules.RuleUtil
Create inference model
jena.ModelFactory
InfModel
Get RDFS reasoner
jena.ReasonerRegistry

Figure 4.5: Query framework

Firstly, all necessary files (the transformed BGT triples and ontology, alignments) are loaded into a RDF database and transformed into jena models (using the ModelFactory/TDBFactory class of the Jena API [4]). For small tests using sample data, the ModelFactory suffices as it stores the data in-memory, usually leading to better performance. The jena models correspond to RDF graphs. Although it is good practice to separate ontology definitions (schema

[4] https://jena.apache.org/documentation/ontology/

information) from instance data, in the test cases this can be stored together in a model for the sake of simplicity.

After generating the RDF graphs from the files, a reasoner must be loaded. The Jena framework provides support for different reasoners:

- **OWL/OWL micro reasoners**: allows reasoning on OWL Lite (a subset of OWL)
- **Generic rule reasoners**: as the name suggests, this reasoner supports user defined rules.
- **RDFS rule reasoners**: implements a subset of RDFS entailments.

A RDFS/OWL lite reasoner provides sufficient support for the problem at hand (which is to attempt to give more meaning to the spatial relation between instances from different sources).

There are two types of queries: simple and complex. Simple queries directly access the RDF dataset, returning only explicit data that was already stored. Complex queries, on the other hand, require rules to be executed and return the inferred information. To enable access to inference (through, for example, SPARQL queries) the data to be queried must be associated with the reasoner – this is done by creating the inference model. Consequently, only complex queries require access to the inference model. Although both simple and complex queries might look the same to the user when typed down, the second type can 'trigger' rules.

Custom rules can be triggered in different ways, depending on the tools and languages that are being used. With SHACL, rules can be executed on a model - which can result in a new model (if using SPARQL CONSTRUCT), a query answer or a warning. The rules can only be applied to *models*, so there is no way to rules that lead to inferred knowledge during query time. To accomplish this, a custom query engine can be created. A query engine is responsible for interpreting the query and coordinating the retrieval of information. A custom engine could be used to identify certain triple patterns or predicates, as to trigger inference during query execution. Regarding SHACL, a proposal for property value rules[5] has been submitted with this idea in mind. The idea is that a SPARQL query could ask for the value(s) of a certain subject's predicate, for example. The same predicate would then be used in a rule definition, making the rule 'fire' whenever such a query is made. In the case of this research, rules can be applied to qualify the spatial relation between instances from two different sources, based on concept alignments. To do so, the type of spatial relation (equals/contained/is-contained) and possible spatial aggregations must be considered first. This is done by calculating the overlap between one or more known features (being queried) and all intersecting features from the second source. This can be done using GEOSPARQL, although an extra user-defined function is needed to calculate the areas of overlap – as GEOSPARQL can only return the polygons of intersections and does not allow metric calculations. After the overlap between the main (known) instance and all other intersecting instances is calculated, rules can be used to decide:

- Whether the overlap between both truly signifies a spatial relation, and is not a consequence of scaling/accuracy discrepancies (this check was also performed during

[5] https://www.topquadrant.com/graphql/values.html

instance-based alignment)

- What type of relation is observed, and between which instances: 1-to-1 (equals/contains/is-contained) or an aggregation (this check was also performed during instance-based alignment)

- Whether these instances share some type of meaning. This is inferred based on the conclusions from 1 and 2 + previously computed ontology alignments. Although this semantic relation only says something about the similarity between the *concepts* used to define the geospatial instances, it can be used to gain more insights into the objects.

4.2.1 Formulation of example queries

Query processing can be rather complex and SHACL/SPIN rules have certain limitations. While it is possible to calculate the area of overlap between instances within a rule itself (in this case, a SHACL rule), SHACL does not promptly support GEOSPARQL/spatial operations. Besides, it is not possible to access remote SPARQL endpoints within SHACL rules. And as mentioned earlier, the proposed extension for property value rules [6] (which allows to create executable predicates) has not been approved yet. This makes it difficult to develop a workflow for automatic query execution. Instead, for the purpose of this exercise, the following is done: intermediary' SPARQL CONSTRUCT queries are created with the Jena API. These are simple and more objective queries made by directly accessing the RDF data (the BGT locally, and TOP10NL via remote SPARQL endpoint), used for calculating overlap between instances. The results of this query are used to trigger a SHACL rule that deals with the more subjective issue: qualifying overlap based on thresholds.

Although the alignments are only partial (because they are based on smaller study areas), the correspondences found between concepts could still be used to improve queries between the two sources. Two examples will be tested, concerning validation and data integration. With regards to validation, a check could be made to see if two 'equal' objects from different sources (with high overlap) are represented by concepts that are marked 'disjoint' in the alignment (meaning they do not often overlap). If this is the case, it could be due to a problem in the classification or due to differences in the update frequencies. Either way, signalizing such differences could be of added value - if, for example, both datasets have recorded different changes throughout their update cycles. To assess whether such queries could be of use, the following is done:

- A SPARQL query is used to retrieve TOP10NL objects within a predetermined bounding box

- For each returned TOP10NL object, a CONSTRUCT query is used to calculate the overlap with nearby BGT objects (a query similar to the one shown in Appendix G) - the results are stored in a Model

- A SHACL rule (similar to the one shown in Appendix H) is used to determine the spatial relation between both objects - the results are again stored in a Model

[6] https://www.topquadrant.com/graphql/values.html

- If the relation is unclear, or it is an aggregation, no further steps can be taken (because these variations could have many causes: related to differences in granularity, geometric changes due to mutations, etc). If, however, the relation is of type 'equal' another SHACL rule (similar to the one shown in Appendix I) is used to find out if the types of both objects are related somehow. If this is not the case, a warning is issued.

The second example involves an integration issue: an end-user might want to retrieve certain object types that could be present in both sources. Even if these object types are expressed through somewhat different aqcuisition rules, an user might consider their similarity to be sufficient, expecting an answer that combines similar information from the two datasets. In the selected datasets, such examples could easily occur with the 'functioneel gebied' objects (parks, camping sites, coomunity gardens - all object types that can be interpreted differently during data collection).

- A SPARQL query is used to retrieve TOP10NL objects (with specific type) within a predetermined bounding box
- The objects are stored in a Model, and a SHACL rule is used to determine if the object type is similar (semantically equivalent/complimentary) to any BGT object
- If this is not the case, only TOP10NL objects are returned. If else, all BGT objects (with similar object type) within the bounding box are retrieved and their ID stored in a list.
- A SHACL rule (similar to the one shown in Appendix H) is used to determine the spatial relation between the TOP10 and BGT objects - the results are stored in another Model
- If there is a clear relation (aggregation, equals, etc), then the identifiers of the BGT objects are stored and removed from the list. All TOP10NL objects and the remaining BGT objects in the list are returned as a result

4.3 Preparation of datasets

After the alignment is finalized and before querying, the data needs to be processed: BGT data has to be transformed into linked data. As a vocabulary for the data model is already available, only the scripts to translate the data to RDF format were needed. The code used was based on the BGTHigh project[7]. The TOP10NL data, on the other hand, can be retrieved from a SPARQL endpoint[8] dynamically through a SPARQL <SERVICE> clause, at query time. Afterwards, the resulting instance data (and ontologies), together with the final alignments and semantic relation definitions can be loaded to a triplestore for querying.

[7] https://github.com/provincieNH/BGTHigh3

[8] https://data.pdok.nl/sparql

4.4 Data collection and processing

In the first phase of the research (see Chapter 3) the data was not required to be in RDF format. Therefore, database files of both BGT and Top10NL are used instead. As large volumes of data are involved and have to be processed during alignment (in the case of instance-based matching), only a subset of the national dataset is used to this end. To avoid sample bias, five different areas within the country are selected for the study using bounding boxes - based on the NUTS-1 country regions[9], as seen in Figure 4.6.

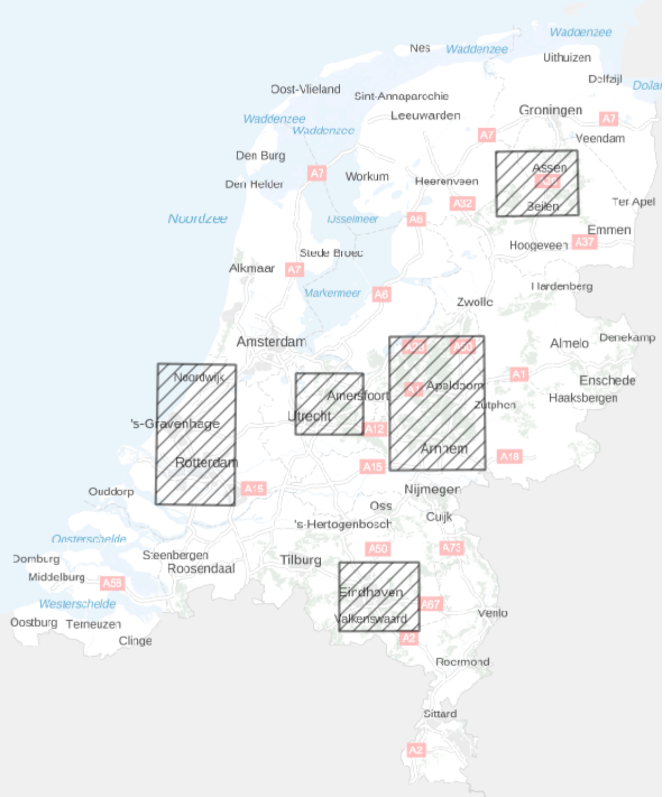


Figure 4.6: Data sampled from BGT/Top10NL

Before selected the bounding boxes, areas were searched and object statistics analyzed based on the presence of different terrain coverages and on types of settlement (rural and urban). This was done to ensure as many different objects were going to be included in the results. To keep the comparisons manageable, only a few object types of both datasets were considered in the analysis: terrain, water and functional areas. These object classes were selected because they represent three very different phenomena, and because they have highly similar attribute values in both datasets. The limitation of this selection approach is that some object types and attribute values will never be present in the sampled data. However, this is unavoidable - and sometimes, certain values are too specific to be encountered frequently throughout the dataset (for example, Bron(Wel) and Kerncentrale in the TOP10NL). In Appendix E and Appendix D a list is given with the frequencies of all attribute values (in relation to the total frequency found in the whole dataset). This also exposes

[9] https://www.regioatlas.nl/kaarten_nuts1regioslandsdelen

the inconsistencies regarding the use of optional features: for example, optional values for building functions in the TOP10NL are not used consistently at all - none of the sampled datasets had any building object with an office or retail function assigned to it, which is clearly incorrect. Nonetheless, a considerable amount of different objects was collected.

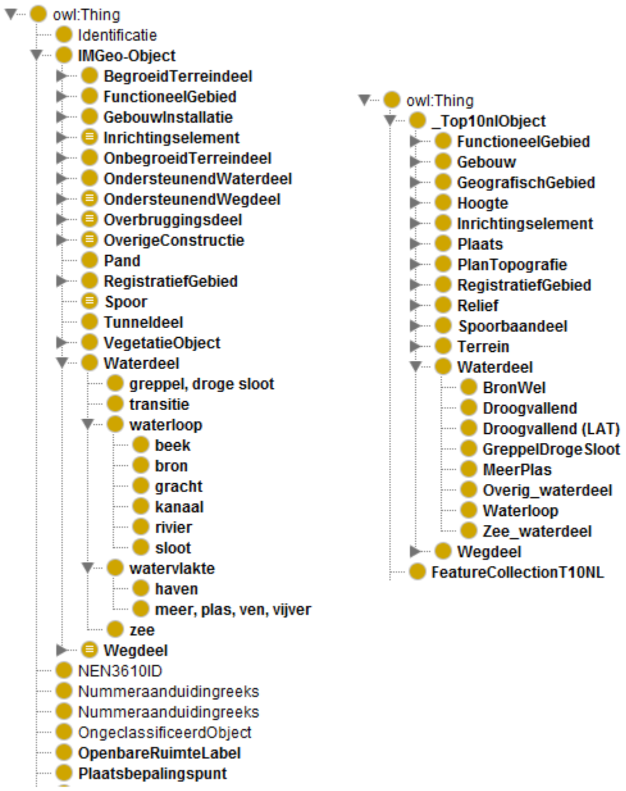
While the TOP10NL and BGT have different update frequencies, the most recent versions available at the time of collection were used. Although these differences in updates could affect the mapping, it is currently not possible to estimate what impact this would have on the results. Still, it should be remembered such differences could lead to misleading correspondences (or hinder correspondences that would otherwise make sense). However, it is not possible to investigate this in more detail, as the BGT does not register the valid time of objects (only the transaction time, when they were entered in the database) and the TOP10NL object contours are often based on preexisting aerial images.

4.4.1 Data transformation

The transformation between BGT records and RDF graphs requires some changes that should be discussed. This must be done as the attribute values in the graphs (at least those of Top10NL) are often modelled as classes if they relate to physical properties of the object in question. This is in line with current efforts to separate between the physical and functional aspects of the main model classes of dutch information models for geo-registries. This trend is also followed when creating the BGT ontology, as it makes the generation of mappings more intuitive (see Figure 4.7). However, some differences between the relational representation and RDF data representation cannot be avoided: codelists cannot be represented, and there is no distinction between properties and relations - in RDF/OWL, both are modelled as 'properties'. Fortunately, the selected data models do not rely much on relations. Finally, the ontology versions of both TOP10 and BGT have been directly linked to the NEN3610 vocabulary found at [10]. This is necessary for the structural alignment. For querying, several random BGT tiles were downloaded from [11] and quickly converted to RDF using the code mentioned in Section 4.3.

[10] https://github.com/Geonovum/NEN3610-Linkeddata/

[11] https://www.pdok.nl/downloads/-/article/basisregistratie-grootschalige-topografie-bgt-#7eedc55878c2562e833f17344aa78cf5



- owl:Thing
  - Identificatie
  - IMGeo-Object
    - BegroeidTerreindeel
    - FunctioneelGebied
    - GebouwInstallatie
    - Inrichtingselement
    - OnbegroeidTerreindeel
    - OndersteunendWaterdeel
    - OndersteunendWegdeel
    - Overbruggingsdeel
    - OverigeConstructie
    - Pand
    - RegistratiefGebied
    - Spoor
    - Tunneldeel
    - VegetatieObject
    - Waterdeel
      - greppel, droge sloot
      - transitie
      - waterloop
        - beek
        - bron
        - gracht
        - kanaal
        - rivier
        - sloot
      - watervlakte
        - haven
        - meer, plas, ven, vijver
      - zee
    - Wegdeel
  - NEN3610ID
  - Nummeraanduidingreeks
  - Nummeraanduidingreeks
  - OngeclassificeerdObject
  - OpenbareRuimteLabel
  - Plaatsbepalingspunt

- owl:Thing
  - _Top10nlObject
    - FunctioneelGebied
    - Gebouw
    - GeografischGebied
    - Hoogte
    - Inrichtingselement
    - Plaats
    - PlanTopografie
    - RegistratiefGebied
    - Relief
    - Spoorbaandeel
    - Terrein
    - Waterdeel
      - BronWel
      - Droogvallend
      - Droogvallend (LAT)
      - GreppelDrogeSloot
      - MeerPlas
      - Overig_waterdeel
      - Waterloop
      - Zee_waterdeel
    - Wegdeel
  - FeatureCollectionT10NL

Figure 4.7: Ontology representation of BGT and TOP10NL

5 Results and discussion

5.1 Ontology alignment

This section discusses the different methods that were used for the alignment, and outlines their shortcomings. As this is not a formal evaluation or benchmarking, no precision/recall is used - we evaluate the results manually. Either way, there is a discussion about the quality of alignment benchmarks - as it is a highly subjective matter. Some of the problems are highlighted below, with examples from the case study.

5.1.1 Terminological matching and limitations

Syntax-based measures work well in cases where related concepts are often similar. The table in Appendix F shows the Jaro-Winkler similarity value for each word pair of the TOP10NL and the BGT - values above 0.7 are marked in green. While the distribution of high similarity values follows the expected results for some classes (’Terrein’ and ’Inrichtingselement’ in both datasets have many words in common), it also shows other patterns. For example, the TOP10NL class ’Functioneel gebied’ seems to match with many BGT classes, and BGT ’Kunstwerk’ has quite some matches in common with TOP10NL ’Inrichtingselement’. Upon closer (manual) inspection many of these matches are inaccurate.

Table 5.1 contains some similarity scores for concepts from the same NEN3610 classes. While there are many similar concepts within the NEN classes (which can easily be identified manually, as well) the limitation of syntax (edit-based distance) methods are clear: it cannot handle small syntactic variances (see ’Duin’ and ’puin’, or ’baak’ and ’bak’). As some classes contain many short-tailed words, this issue is even more evident in scores between different classes (see the last few rows of Table 5.2). Nonetheless, most scores (apart from the cases just mentioned) seem to reflect word similarity more appropriately when higher thresholds are maintained. Still, another problem is the presence of fairly similar words that, in certain contexts, could be considered to be opposed. This is the case with ’hoogspanningsmast’ and ’laagspanningsmast’: both represent the same type of object (a tall pole), but have different workings (one works on high voltage, the other on low voltage). In this particular case, the BGT ’laagspanningsmast’ is rdfs:subClassOf BGT ’Mast’, so TOP10NL ’hoogspanningsmast’ could be related to this more general concepts. However, these types of decisions would have to be made manually - there is no way of telling automatically that these two concepts should be subsumed by a more general concept instead of sharing a direct link. These types of problem also persist when using other linguistic methods (such as analyzing descriptions), because they are related to one’s interpretation. Moreover, other datasets might not have a more general class such as ’Mast’ present, in which case this would have to be created in a shared vocabulary.

	Same NEN classes			
	Class	TOP10NL concept	BGT concept	Similarity score
Reliable	Water	Waterloop	waterloop	0.93
	FunctioneelGebied	Volkstuinen	volkstuin	0.84
	Terrein	Zand	zand	0.83
	Terrein	Waterloop	waterloop	0.93
Unreliable	FunctioneelGebied	Heemtuin	speeltuin	0.78
	Terrein	Duin	puin	0.83
	Inrichtingselement	Hoogspanningsmast	laagspanningsmast	0.83
	Inrichtingselement	Baak	bak	0.92

Table 5.1: Syntax-based similarity scores between concepts from the same NEN classes

	Different NEN classes			
	Classes	TOP10NL concept	BGT concept	Similarity score
Reliable	FunctioneelGebied/GeoObject	Windturbinepark	windturbine	0.86
	Gebouw/GeoObject	Windturbine	windturbine	0.94
	Terrein/Weg	Zand	zand	0.83
	Inrichtingselement/Kunstwerk	Hoogspanningsmast	hoogspanningsmast	0.96
Unreliable	FunctioneelGebied/Inrichtingselement	Waterkering	wegmarkering	0.82
	Gebouw/Inrichtingselement	Kasteel	Kast	0.91
	Gebouw/Inrichtingselement	Fabriek	abri	0.86
	Terrein/Weg	Duin	puin	0.83

Table 5.2: Syntax-based similarity scores between concepts from different NEN classes

In Table 5.2 another issue is highlighted as well. While the BGT has one object of type ′windturbine′, the TOP10NL dataset also contains an object of type ′Windturbinepark′ - which represents a collection of windturbines. One could image a situation in which ′windturbine′ (or other similar objects) from one dataset are linked to the ′park′ represented in another - if the windturbine objects have more information attached/are more recent and if the park object represents official boundaries, for example. However, this is not the case here. In the TOP10NL, ′windturbinepark′ represents an object type that is based on other, more detailed, objects. The geometry has no relation to any administrative boundaries, but is rather the result of buffer area calculations (as can be seen in Figure 5.1). Therefore, linking such concepts directly is unnecessary - again, a choice that can only be made through manual inspection.

Another issue is the use of very generic concepts to represent objects - leading the classes ′Terrein′ and ′Weg′ to share the concept ′zand′. Although the concept is probably the same for both classes, it is used differently. In the BGT (and also in the TOP10NL), ′Weg′ or road objects are functional by nature - a road is made to support displacement of some sort. The attribute value ′zand′ then describes an additional (physical) property of a road object. Regarding the ′Terrein′ objects, many subtypes are already physical in nature and relate to landcover types: gras, sand, trees, bushes. Hence, a ′sand terrain′ might be different from a ′pedestrian road made of sand′. The degree to which these things should be different is hard to quantify - but taxonomic and extensional matching could help dissociate both.

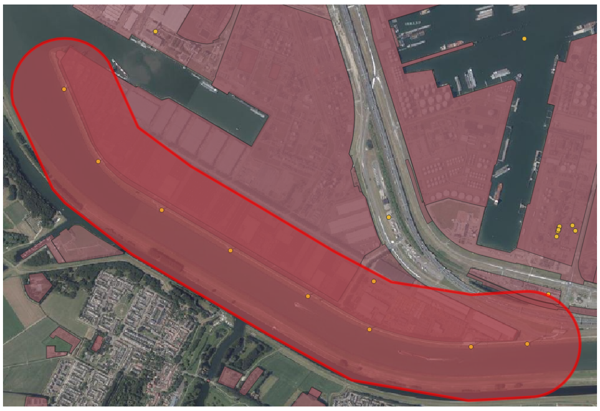


Figure 5.1: TOP10NL representation of Windturbinepark Hertelbrug II in Rotterdam in dark red (points are the windturbine object - both in TOP10NL and BGT

TOP10NL	BGT	Syntax similarity	Syntax + structural similarity
Terrein Grasland	Terrein grasland agrarisch	0.75	0.84
Terrein Grasland	Terrein grasland overig	0.78	0.86
Terrein BosNaaldbos	Terrein naaldbos	0.84	0.92
FunctioneelGebied Waterkering	Inrichtingselement wegmarkering	0.82	0.75
Inrichtingselement Scheepvaartlicht	Inrichtingselement scheepvaartbord	0.76	0.83
Gebouw Kasteel	Inrichtingselement Kast	0.91	0.82

Table 5.3: The similarity before and after using the common path between concept classes (with weigh = 2 or weight = 10, if initial similarity > 0.74).

5.1.2 Structural matching

Matching should been performed based on typical characteristics of an object - things that make these objects distinct, which is most often their ’type’. In the case of the BGT and TOP10NL, these ’types’ are represented using rdfs:subClassOf - following a taxonomic structure. As such, linking the abstract object classes to their common denominator (the NEN3610 class primitives) is a straightforward way to include measures of relatedness when calculating similarity. Similarity values for two concepts are assigned higher weight depending on the path length that connects them to a common NEN3610 class. This process corrects some of the shortcomings from syntax-based matching - depending on the thresholds that are used. Although the formula introduced in the previous chapter was employed (were the similarity with parent nodes counts as well), improvements to the resulting similarities were only noted after substantially increasing the weight used - when the initial similarity between concepts is already above a threshold. This means the relatedness between concepts (whether they belong to the same NEN class) can boost the similarity value when it is already high. In Table 5.3 some differences are shown between a purely syntactic and structural approach: the similarity values for similar concepts from the same classes are increased substantially. However, the syntax problem still prevails: words that are most likely related to different things depending on the context (such as ’scheepsvaartlicht’ and ’scheepsvaartbord’) are marked as more similar. Besides, unrelated (but very similar) words will still score high compared to others (such as ’kast’ and ’Kasteel’).

Regarding this type of matching, it might have been just as effective to directly multiply the syntactic similarity with a fixed factor based on whether the NEN3610 class connecting both classes was the same. However, the choice of which factor to use would still be subjective - and it does not solve problems introduced by syntactic matching. Moreover, the TOP10NL ontology models some 'typing' attribute values as SKOS:concepts instead of rdfs:subClassOf. In this case, both modeling techniques were regarded having equal importance when calculating similarities. However, giving similarity calculations between rdfs:subClassOf and SKOS:concepts less weight could be useful for avoiding situations such as the one described in Section 5.1.1 - the same word 'zand' describing different aspects of different objects types. One could say that if a value is not modelled as 'typing' (rdfs:subClassOF), then it represents aspects that are less important. However, such an idea would only hold for the datasets in question - because they were modeled with this in mind.

Besides, defining the degree to which these 'secondary' attributes are similar to one another cannot be done with only syntactic/structural approaches. The type of matching presented above is more appropriate for singling out very similar/identical words: syntax is used to identify similar words, and taxonomic relations to incorporate relatedness (used to improve the similarity score between words that are already considered relatively similar). extensional matching can be used in order to determine how exactly these concepts are related to each other.

5.1.3 Extensional matching

While syntactic and structural information can be used to find similar words, extensional matching could help refine the relations between concepts. A simple example would be: if two concepts are very similar and their instances always overlap, then these concepts are equivalent to one another. In the case of linked data, it would then be possible to express one dataset in terms of another. Unfortunately, when performing extensional matching on data originating from different scales/granularities this will most likely not be the case. The results from the extensional matching (in Table 5.4) support this claim: between 10-20% of TOP10NL Terrein objects do not seem to map to any BGT Terrein object. Similarily, between 10-25% of BGT Terrein objects do not relate to any TOP10NL Terrein object. As expected, most relations between objects of the class NEN3610 Terrein are aggregations between one TOP10NL object and multiple BGT objects. Between the NEN3610 Water classes the majority of BGT objects do not map to any TOP10NL object because certain water objects can be represented as lines depending on their width. Linestrings were not matched in this process, as it would often require buffers and make computation much slower - and there are not many types of water objects to be matched. Surprisingly, between TOP10NL Gebouw and BGT Pand around 50-65% of matches are of type 'equals' - meaning most of the buildings in the selected area are represented with the same granularity (despite the differences in scale between both datasets).

	TOP10 Terrein - BGT BegroeidTerreindeel			BGT OnbegroeidTerreindeel			TOP10 Water - BGT Waterdeel		
	B1	B3	B4	B1	B3	B4	B1	B3	B4
no mappings									
% TOP10 objects	40.71339	42.06637	39.66944341	65.11768	77.10259	69.92094	44.10094	40.23643	44.2909
% BGT objects	24.47216	19.69037	23.55278067	10.98783	9.610571	9.995352	84.9869	93.72748	94.66123
equals’ mappings									
% TOP10 objects	20.34204	29.40087	22.55929326	5.313755	9.290948	8.693691	31.06091	44.33012	40.69788
% BGT objects	5.722317	17.9873	7.64463092	2.103798	12.37999	6.976687	7.574646	4.044398	3.891762
top10 within bgt									
% TOP10 objects	0.074738	0.033308	0.055711784	0.411794	0.699475	0.693744	2.246003	2.211033	2.297461
% BGT objects	0.021238	0.020488	0.019112715	0.187137	1.064673	0.639693	0.593679	0.207986	0.23159
bgt within top10									
% TOP10 objects	6.497846	6.00383	6.410834616	6.530086	4.084437	5.78872	2.232792	2.495622	2.314735
% BGT objects	1.8465	3.692985	2.199327414	2.967561	6.216927	5.337704	0.590187	0.234756	0.233332
top10 aggregates many bgt									
% TOP10 objects	29.89097	19.1294	29.01124847	18.74469	5.889333	12.4224	6.156692	3.436953	2.470202
% BGT objects	67.85116	58.41166	66.46992705	83.39471	69.08647	76.21395	5.210407	1.573279	0.750492
bgt aggregates many top10									
% TOP10 objects	2.662739	3.616038	2.496418528	4.304053	3.518195	2.951398	16.07874	8.253065	9.414407
% BGT objects	0.08662	0.197198	0.114221225	0.358958	1.64137	0.836615	1.044177	0.212104	0.23159

Table 5.4: Percentage of matching objects between TOP10NL and BGT classes, according to the different topological relations

With regards to the resulting alignments: from 2440 correspondences, only 15 had 'semantic complimentary' or 'semantic equivalent' relation types. Over 2000 bridging rules had 'no support', meaning the overlap between objects might not be related to its semantics. Below are some of the results - as expected, Terrein types 'Heide' and 'Grasland agrarisch' are related. Still, many object types that looked similar, are actually not that related in the dataset. All Terrein correspondences show an asymmetric relation from the BGT object types to the TOP10NL object types. For example: the concept of BGT Heide is subsumed TOP10NL Heide, with 93%. confidence. The fourth rule (rule 449) has a much lower measure of confidence attached to it. The reason is that, while 'Bunkers' probably often contained 'Groenvoorziening' geometries, many 'Bunker' objects were also found in other types of relations (aggregations, for example). However, this specific case is still interesting as it could actually represent complementary information - is there a reason why 'Groenvoorzieningen' are often found inside bunker geometries? When looking at the data, it appears one of the study areas (B3) has a military facility that houses numerous bunkers (see Figure 5.2. This probably skewed the numbers - the association between the concepts only holds on that specific location.

When analyzing the alignments for each city separately, the only different bridging rule found established a connection between TOP10NL Naaldbos and BGT Naaldbos - found in study areas B1, B3 and B4. This could indicate differences in interpretation by the stakeholders, but nothing should be ruled out. Finally, it was mentioned that some TOP10NL SKOS:member listed attribute values should be taken into consideration during extensional matching as they could expose more specific alignments. However, many of these attributes are optional (see the list in Annex X) and did not occur often enough in the matched geometries lists to be taken into account.

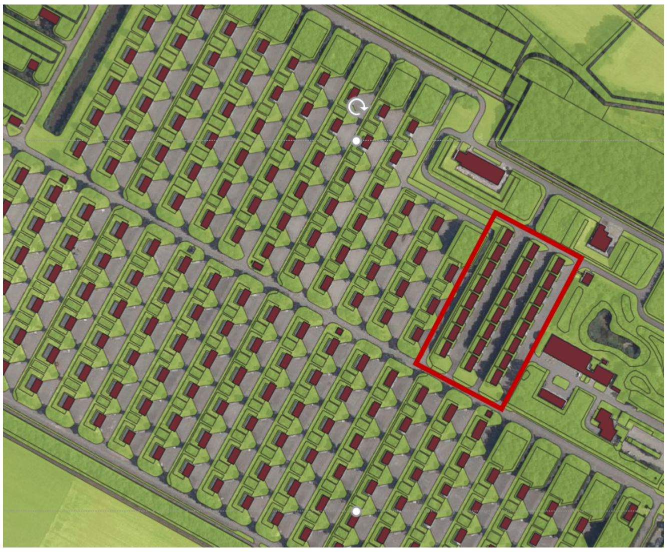


Figure 5.2: TOP10NL Bunkers and BGT Groenvoorziening: military facilitay with many objects counted

```
link_bgt_top10:rule1947 a align:Cell ;
    align:entity1 top10:Heide ;
    align:entity2 bgt:Heide_BegroeidTerreindeel ;
    align:relation rel:sem_equivalent_bgt_to_top10 ;
    align:measure 0.9354095588020879
    dc:subject align:equals .

link_bgt_top10:rule1976 a align:Cell ;
    align:entity1 top10Grasland ;
    align:entity2 bgt:GraslandAgrarisch_BegroeidTerreindeel ;
    align:relation rel:sem_equivalent_bgt_to_top10 ;
    align:measure 0.9842482527745624
    dc:subject align:equals .

link_bgt_top10:rule2311 a align:Cell ;
    align:entity1 top10:Duin ;
    align:entity2 bgt:GeslotenDuinvegetatie_BegroeidTerreindeel ;
    align:relation rel:sem_comp_bgt_to_top10 ;
    align:measure 0.9080882352941178
    dc:subject align:equals .

link_bgt_top10:rule449 a align:Cell ;
    align:entity1 top10:Bunker ;
    align:entity2 bgt:Groenvoorziening_BegroeidTerreindeel ;
    align:relation rel:sem_comp_top10_to_bgt ;
    align:measure 0.5952380952380952
    dc:subject align:top10_contains .
```

Listing 5.1: Some bridging rules from the matching process

5.2 Query and reasoning

The alignment discussed in the previous sections was made to explore relations between objects and concepts from both datasets. As such, it did not rely on semantic web technologies - although the structure of the ontologies was used to measure the similarity between concepts. Geo-ontology alignment can be done through extensional matching (using the geographic overlap between objects, a way of data interlinking), as was done here. On the other hand, data interlinking itself can also take advantage of ontology matching. The main idea behind a query for data validation described in Chapter 4 was to spot unlikely associations between object types, so these could be investigated in more depth. An initial query was then used to find the intersections between a certain TOP10NL instance and overlapping BGT instances. The output graph of these queries often revealed big differences in granularity: many very small BGT objects were part of one TOP10NL object. This can be seen in the listing below ('mainRelativeArea' is the intersection geometry in relation to the TOP10NL object, and 'secRelativeArea' in relation to the BGT object geometry). Finally, when applying the constraint to ensure the TOP10NL and BGT classes within this aggregation are not disjoint (according to the alignment), no warnings were encountered. This was the case for over 200 TOP10NL instances tested for intersections with the BGT instances from a randomly chosen location. One reason for this could be that in the alignment there are very little unambiguous correspondences (such as disjoint, equivalent or complimentary). Although there are a few hundred correspondences of type 'disjoint', these are not that common when looking at the whole. So, the chance of catching overlapping features with 'disjoint' types that should not relate is small. Moreover, as a few study areas were used for the alignment, the 'disjoint' relations might only reflect local problems.

```
maps:relation_obj1   a      maps:OverlapRelations ;
maps:hasOverlap    [ a bgt:Groenvoorziening_BegroeidTerreindeel ;
    maps:mainRelativeArea  "0.3858979";
    maps:secRelativeArea   "100.0" ;
    maps:secfeature        <http://bgt/id/Groenvoorziening/G0150.2a6f...>
  ] ;
maps:hasOverlap    [ a bgt:BegroeidTerreindeel ;
    maps:mainRelativeArea  "0.25251818" ;
    maps:secRelativeArea   "100.0"< ;
    maps:secfeature        <http://bgt/id/Groenvoorziening/G0150.a8e8...>
  ] ;

  maps:mainfeature   <http://brt/top10nl/id/terrein/130511636> ;
       maps:maintype      top10:Terrein ,   top10:Grasland
```

The second query type was aimed at exposing the relations between objects of different sources. The example in Figure 5.3 shows two TOP10NL SportComplexSportTerrein objects and multiple BGT Sportterrein objects. As both object types are often seen together, one could request all objects from both sources to gain a general sense of the location of sport facilities. To avoid duplications, the spatial relation between overlapping objects can be used in the result. The SPARQL query and SHACL rules are used to signalize that the two TOP10NL objects aggregate many BGT objects - and the smaller BGT objects outside of the large complexes can be returned as well. In this case, there were 17 BGT objects in

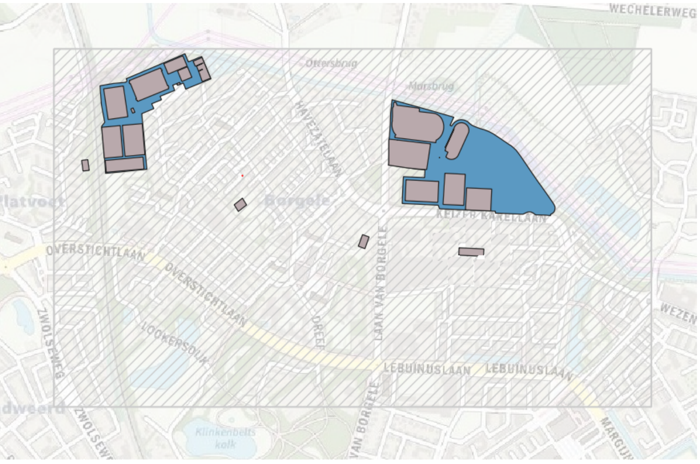


Figure 5.3: Relation between BGT objects (light grey) and TOP10NL objects (blue) used in query

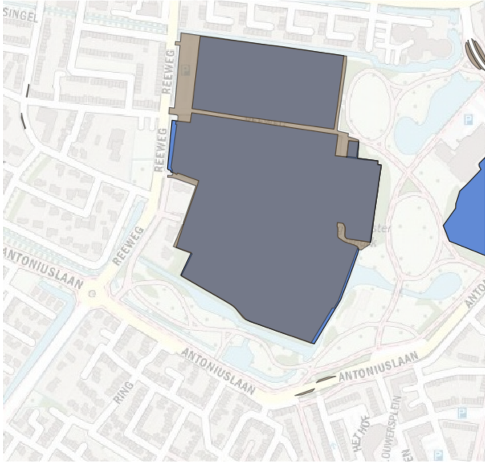
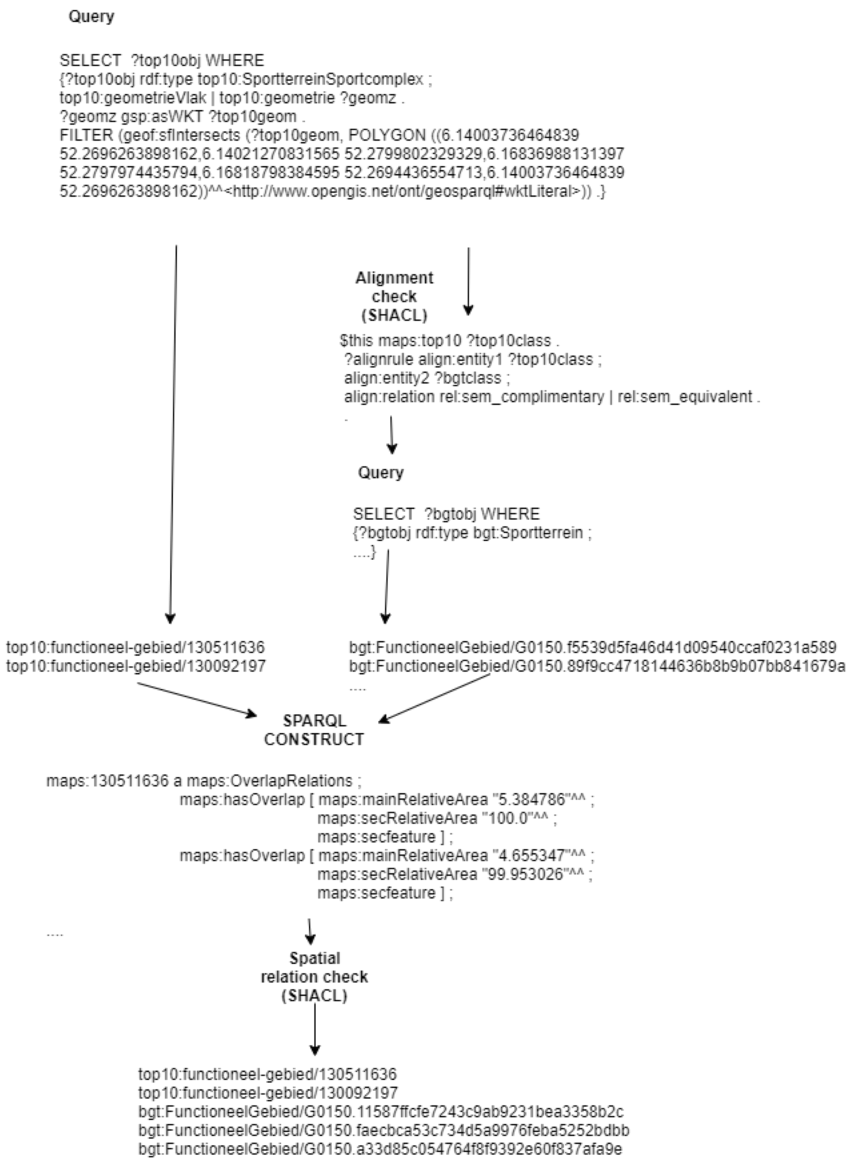


Figure 5.4: Difference in granularity: BGT object (light grey) is coarser compared to TOP10NL object (blue)

total, from which 14 were aggregated inside TOP10NL objects. While it might seem like this aggregation results from scale differences alone, most of these TOP10NL objects actually have names attached to them - in the figure, for example, complexes Keizerslanden and Borgele are represented. This information could be linked to the BGT objects. More importantly, although the BGT is more detailed, some of its objects are represented in coarser detail - this could also be the case with other datasets. Such a situation can be seen in Figure 5.4, were the light contour represents a BGT Sportterrein, while the TOP10NL SportComplex below it seems slightly more detailed. These types of variations within datasets must be reconciled before combining information - as shown in the query example in Figure 5.5, the geospatial overlap and semantic similarity can help determine how this is done. However, it is difficult to determine the correct way in which to display this knowledge because it depends on the demands of the user requesting it.



```
SELECT  ?top10obj WHERE
{?top10obj rdf:type top10:SportterreinSportcomplex ;
top10:geometrieVlak | top10:geometrie ?geomz .
?geomz gsp:asWKT ?top10geom .
FILTER (geof:sfIntersects (?top10geom, POLYGON ((6.14003736464839
52.2696263898162,6.14021270831565 52.2799802329329,6.16836988131397
52.2797974435794,6.16818798384595 52.2694436554713,6.14003736464839
52.2696263898162))^^<http://www.opengis.net/ont/geosparql#wktLiteral>)) .}
```

```
$this maps:top10 ?top10class .
 ?alignrule align:entity1 ?top10class ;
 align:entity2 ?bgtclass ;
 align:relation rel:sem_complimentary | rel:sem_equivalent .
 .
```

```
SELECT  ?bgtobj WHERE
{?bgtobj rdf:type bgt:Sportterrein ;
....}
```

```
top10:functioneel-gebied/130511636
top10:functioneel-gebied/130092197
```

```
bgt:FunctioneelGebied/G0150.f5539d5fa46d41d09540ccaf0231a589
bgt:FunctioneelGebied/G0150.89f9cc4718144636b8b9b07bb841679a
....
```

```
maps:130511636 a maps:OverlapRelations ;
                maps:hasOverlap [ maps:mainRelativeArea "5.384786"^^ ;
                                  maps:secRelativeArea "100.0"^^ ;
                                  maps:secfeature ] ;
                maps:hasOverlap [ maps:mainRelativeArea "4.655347"^^ ;
                                  maps:secRelativeArea "99.953026"^^ ;
                                  maps:secfeature ] ;

....
```

```
top10:functioneel-gebied/130511636
top10:functioneel-gebied/130092197
bgt:FunctioneelGebied/G0150.11587ffcfe7243c9ab9231bea3358b2c
bgt:FunctioneelGebied/G0150.faecbca53c734d5a9976feba5252bdbb
bgt:FunctioneelGebied/G0150.a33d85c054764f8f9392e60f837afa9e
```

Figure 5.5: Steps and result fragments for one query combining TOP10NL and BGT objects

5.3 Discussion

If user needs are clearly defined (within the scope of an application, for example), it should be possible to devise a coherent system - or a set of rules - that takes advantage of different information sources. However, each application or use case could require or benefit from different (custom) solutions, that are not directly supported by specific tools or languages. Users, in this case, will most often be data users - people responsible for building applications and providing services to end users. Although RDFS/OWL is useful for representing taxonomic relations, there are confusing situations as seen with the selected datasets and their ontologies: some properties were represented as taxonomic, while others were constructed with OWL:Property. And although the taxonomic subclasses are supposed to represent *types* (setting them apart from other properties), the interweaving of physical and functional attribute values makes it difficult to separate them.

It was also difficult to set apart the technologies, as they can be used differently depending on the application. For example: SWRL is a rule language for inference, and can be used for mappings; SHACL was designed primarily for validation, but now has extensions that allow rule definitions (which can in turn be used for mapping). This is partly the nature of the problem: the specifications evolve as the needs are understood better; but it also makes it more difficult to grasp the topic and all possibilities. Some seem to suggest these complex logics are not the best for representing things on the web (geo or not). This warrants some reflection whether or not the semantic web technologies in their current form/extent are actually the best solution for exposing data from geo-registries. Linked data is only one way of publishing data on the web. And while linked data standards are becoming less complex, more accessible and more user friendly (such as SHACL and JSON-LD), applying best practices as explained by Brink [2018] can be done by different means.

With regards to the usability of different matching strategies: using thesauri or concept descriptions, while useful, might lead to results which are not in line with the structure of the ontologies. Čerba and Jedlička [2016] found that when basing similarity scores of geospatial concepts on their descriptions, the scores where often lower than specialists would consider them to be. They conclude that the semantic relations between concepts are created based on implicit semantics, which involves subjective views of those in charge of the ontology creation (see again the meaning triangle explained in Sowa [2000], or the concept of pragmatic heterogeneity). This issue also involves the discussion surrounding similarity and relatedness - the second one is very common with geospatial data. This will also be the case with the registries in this use case, and will also apply to other geo-registries - as they are created and maintained in similar ways. Therefore, instance-based methods are necessary, above all. Syntax-based methods, while not that reliable in this case, might be used (together with high thresholds) to single out words that are basically the same across the ontology terms. Care should be given to the fact that such methods might perform differently depending on the structure of the language in question - as was the case here with many unrelated Dutch words being considered similar, such as 'bak' and 'baak'.

Regarding ontology alignments as a whole, even if there is ultimately a 'right' matching method for registries (and applications of registry data), it would still be necessary to evaluate the performance - by using benchmarks. As mentioned in Chapter 2, OAEI plays an active role in this. Unfortunately, there is a lack of geospatial benchmarks, and most of the initiatives focus establishing equivalence relations(owl:sameAs) between sources. Also, the quality of the reference alignments used in the benchmarks has been questioned by several authors (mentioned in Zhou et al., 2019) - which again relates to the subjectivity involved in geospatial ontology creation.

The approach used to create geospatial ontologies (underlying data model) also influences the results. Certain attributes with variable meaning (such as high-rise/low-rise) could be easily deduced from features such as 'height'. It is probably best to do so, as height classes could be dependent on certain context. Even more, there does not seem to be a clear or uniform way to express alignments (of course this has to do with the many different types that can be constructed) or how to express their quality: While traditional geospatial ontologies (i.e. registries) are defined in a top-down approach, researchers stress the importance of considering bottom-up approaches for ontology enrichment (Kokla and Guilbert [2020]; Kokla et al. [2018]; Janowicz et al. [2012];). Such initiatives could allow incorporating perspectives from different users - and different contexts. This goes hand in hand with the idea that ontology standardization might be more difficult to achieve than was expected. Janowicz et al. [2012] suggests that instead of standardizing ontology representations, responsible bodies could standardize the process of alignment. This would shift the focus of research from developing interoperability to avoiding incompatibility. When it comes to geo-registries (which are usually already part of a larger system), this might be an interesting approach to explore.

Finally, it should be remembered this research also explored data interlinking besides matching, to improve interoperability. The query framework concerned this interlinking, while the matching led to some (very little) alignments - both make use of instance data. According to Euzenat and Shvaiko [2013]: "Data interlinking and ontology matching can be seen as dual operations. On the one hand, when confronted with two data sets using different ontologies, data interlinking can take advantage of ontology matching. On the other hand, ontology matching can implement extensional methods, or instance-based methods, which may take advantage of data interlinking . Hence, the two processes may be used for reinforcing each other." (page 13). This duality is especially relevant for objects from geo-registries, as they already contain links in the form of spatial relations.

6 Conclusions and recommendations

6.1 Conclusions

One of the main purposes of open government initiatives is to provide users with high-quality data from different sources. Initially these users were mostly other governmental bodies, but nowadays many believe such data should be (at least partially) available for a larger audience – citizens and businesses, which is in tune with the 'open' data movement. Linked data and it four principles came to be seen as a means to achieve this. Regarding geospatial government data published using these linked data principles, the linkage between heterogeneous sources is usually missing. For geospatial data such links are even more relevant because the datasets represent different views on a same location – integrating these views could help us understand the different ways in which we interpret reality. One could say there is already a link due to the geospatial location of objects. However, qualifying those links and understanding the relation between spatially linked objects is more difficult. Therefore, the main aim of this research was to investigate how linkable geospatial linked data actually is, by answering the following question: "To what extent can ontology-based solutions using semantic web technologies, contribute to integration and use of data from geospatial registries?"

To answer the main question, three sub-questions were formulated. These sub-questions were explored in a qualitative study to understand the theory behind semantic web technologies (main tools, languages and related research) and to explore the usefulness of resulting 'linked data'; and a quantitative study to understand the relation between data and concepts from different sources. To this end, two Dutch geo-registries (BGT and TOP10NL) were used as a case study. The limitations related to the use of a case study for answering the questions have been examined in the discussion chapter. Below, each sub-question will be answered separately.

The first sub-question was: "What type of ontology-based techniques are best suited in the case of integrating data from the geo-registries?". The main assumption behind this question is the existence of certain methods that are more applicable to geo-registries. Through the case study and a discussion of geo-registries in general, it can be said these datasets do indeed share similar structures: they are based on standard data models, often make use of predetermined attribute values, and contain large volumes of geospatial data (which could include temporal aspects). Although there is no all-encompassing strategy that applies to geo-registries, instance-based methods must always be a part of the solution. This seems logical, as there is already an explicit spatial link between data from different geo-registries (this is something not all sources that need to be integrated can rely on). However, in the case of geospatial data it is even more relevant due to context and relatedness carrying much more weight in the geospatial domain than similarity of words. This was verified by the fact that the selected datasets contained many similar words that did not overlap significantly in the data. Often, this was a result of different 'acquisition rules' - which dictate how the concepts of the data models ought to be interpreted. As such, it is not possible to fully align both datasets - the words used in the models simply describe different things. At best, rules can be created to dictate when overlapping objects from different sources belong to the same 'ensemble' based on their geometry and attributes. The idea of using syntax and structural matching was also to expose some of the limitations that should be expected if using such methods automatically - they provide insight on the importance of *relatedness*). Based on the literature study, geospatial interlinking could profit from a structural strategy based on Tversky's model: if two object classes share many characteristic features they could be considered more similar. However, the data models from the case study were not suitable for such a strategy. Therefore it can be said only instance-based methods are suitable for integrating data from geo-registries, as expected. While the use of other strategies could provide new insights with regards to data integration, they do not present a definite solution.

Regarding languages and tools, semantic web models are usually described using OWL – which does not support geospatial data types. Luckily, the GeoSPARQL extension is well supported within RDF triplestores and allows interpreting the geometry and spatial relations between instances of ontologies. To enable better geospatial relations between instances from different sources, these relations could be qualified rules. However, even languages that do allow such constraints to be made (such as SHACL) cannot deal with federated queries directly. Therefore, a lot of custom handling is involved if one wants to calculate relations based on location during queries. While not all classes of the data sets were analysed, most of them seem to have a 'typing' attribute and some other attributes characterizing other physical or functional aspects. The 'type' of objects are easily represented using RDFS. Even if OWL supported geospatial data, the models from the case study would not necessarily profit from the capabilities of OWL. Although this might not be true for all geo-registries, it is something worth considering.

The second sub-question was "How does the overlap and differences between the data (models) affect the correspondences between the data sources?". To answer this question geometric representation, semantic representation, and granularity were observed. It was assumed the BGT contained more detailed information, and overall this is definitely the case . As such, it was also expected most TOP10NL instances would either aggregate BGT instances or they would be equal to one another at best. The results of spatial overlay used for the alignment partly confirmed this, but it also revealed BGT objects can aggregate many TOP10NL objects (this is especially the case with terrain/gras types). Thus, while the BGT is based on a larger scale and overall has polygons represented at a finer granularity, this is not always more detailed. The same could hold for other datasets.

Considering geometric differences, it was known beforehand that certain class objects in the TOP10NL can be represented using different geometry types. In this research, only overlapping polygons were considered as this concerns most of the objects. In cases were different geometry types are used it is not possible to establish spatial links, which hinders the alignment - this was the case with the 'Water' class of the selected datasets. Looking at the dataset revealed that even the use of a buffer was not sufficient, as the representation itself is not consistent.The degree to which this affects alignments of other datasets cannot be estimated.

Regarding the semantics of the objects, while there seems to be more definite relations between some concepts in both datasets, the findings suggest that the object types attached to the geometric instances could be interpreted differently depending on the region. This is possibly due to the ambiguity in the definition of acquisition rules. However, this cannot be confirmed as it could be the result of differences in update frequencies, for example. The research did not consider possible mutations of BGT objects and how this might affect matching. Therefore, any other similar efforts should consider the impact of temporal aspects when defining such relations.

Because of the approximate nature of matching (based on the geospatial instance data), registering specific information on the quality of alignments is important. There does not seem to be a standard way of doing this. Even with current application agnostic alignment formats (such as EDOAL), there is still a problem with defining measure of reliability of alignments. This is important in geospatial instance-based matching, as there are many decisions that could affect the results - from thresholds used in spatial comparisons to differences in update frequencies of the data sources that have to be mediated in order to create correspondences.

The final sub-question was: "What is the added value of having custom semantic relations incorporated into ontologies?". This is answered by evaluating (through some queries) whether ontology matching alone generates useful correspondences, or if rule-based reasoning is advantageous. Regarding the results of the alignment, not many semantic relations were found. Besides, the established correspondences were fairly simple - between two classes, without any transformations involving properties. Even if this was not the case, it would not change the problem of subjectivity involved when producing alignments based on similarity-based matching. The alignment was based on many thresholds and subjective similarity metrics and there is no standard way to represent this on the semantic web, as mentioned earlier. More importantly, geo-registers are usually domain ontologies that interpret similar symbols or words as very different conceptualizations of a reality. As such, resources from such datasets are better integrated on the instance level, through their spatial relations directly, and using rule-based reasoning - as was illustrated with the queries.

Reflecting on the research, the experimental nature of the products stands out. This is largely to the multitude of tools and languages that could be employed within the semantic web context. There is not a ultimate solution that can be translated in an application - the process is much more complex and involves organizational as well as technical issues. Although ontology and semantic engineering are fields on their own, research on *geosemantics* presented in this thesis shows how the geospatial domain has been applying this knowledge to its advantage. Access to comprehensive and complete geospatial data services is of great importance - Spatial Data Infrastructures attest to this. While linked data might still present many challenges (both from a technical and organizational point of view), the search for better semantic integration of existing geospatially enabled data will continue to develop itself.

6.2 Recommendations & Future work

Due to time and resource constraints, many interesting questions and topics could not be explored. Below, some of these are explained and recommendations for future work are given.

- Temporal aspects of data objects were out of scope for this research. However, such aspects could be (partly) responsible for the lack of correspondences between concepts which were thought to be similar on an instance-level. Considering such aspects during instance-based matching could shed more light on this issue. This could be done by looking if the dates between the registration of two objects from different sources (or their mutations) differ too much. If this is the case, the match becomes less reliable. Moreover, it is important to examine the coherence between the registration of an object in the database and the time change was observed physically.

- While some research in the related section has pointed out the importance of context, this idea should actually be central to any geospatial ontology integration problem. This is the case as integration should serve a purpose, and the proper metrics to express similarity and relatedness will very much depend on this purpose - thus, it should be researched how these metrics can be made adjustable and if this can be done in an application agnostic way. This involves a better understanding of the data users as well. Techniques related to user modeling and adaptive systems could be examined in more detail.

- In the selected datasets, differences in granularity were observed - even if the BGT is generally more detailed. Although it was not investigated, this could be caused by differences in how data owners are able to carry their tasks. In such a case, the linked data approach could be advantageous due to its decentralized nature: data access could be handled by (the larger) data owners. This could, for example, encourage publication of more up-to-date information - directly from the source. More research on the in-house practices of data owners is needed for this.

- An issue that inhibited the work was the interweaving of conceptual/encoding problems due to OWL and linked data not separating these two facets of information integration. This brings the question whether standardizing ontological encodings is the way to go - maybe one should rather standardize alignments. The ways in which this could be done requires further research.

- Regarding SHACL, it should be interesting to investigate the use of SERVICE inside SHACL rules (especially inside sh:values declarations). In such a way, queries containing certain predicates could trigger calls to other federated sources within the system. This would means users do not need to compose all spatial queries themselves to get additional information on a certain object.

A BGT data model

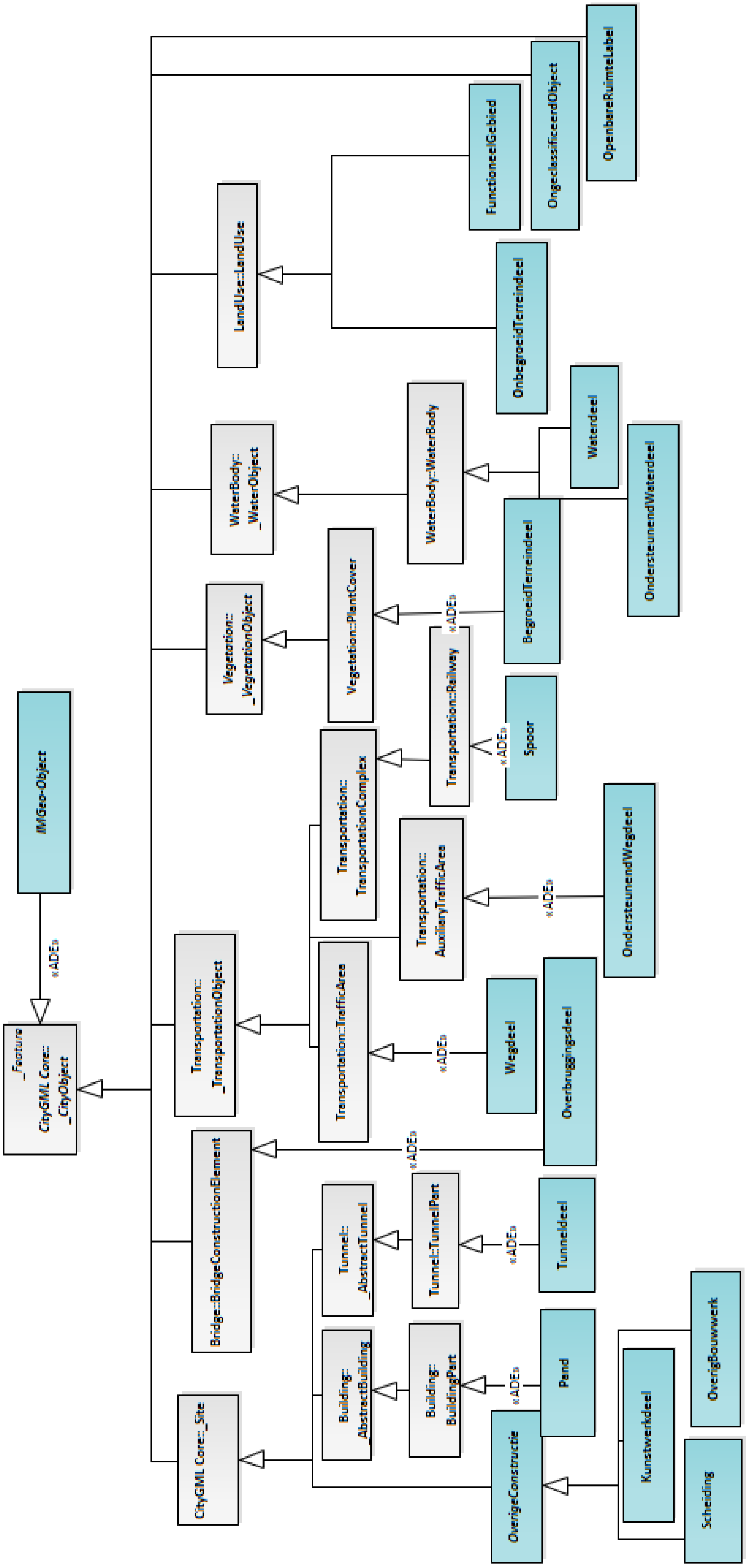
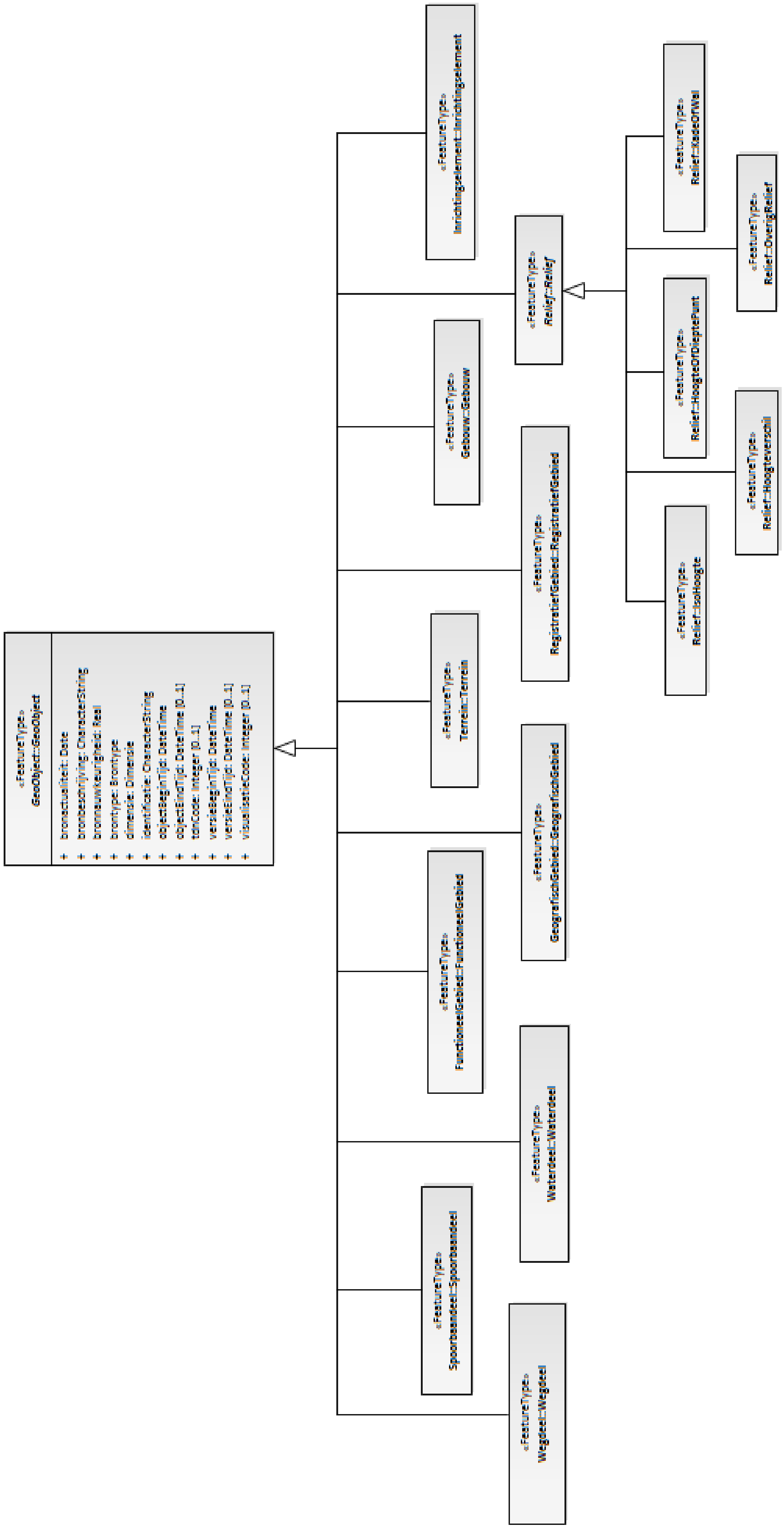


Figure A.1: Diagram overview of the IMGEO data model, used for the BGT dataset

B Top10NL data model


«FeatureType»
GeoObject::GeoObject
+ bronactualiteit: Date
+ bronbeschrijving: CharacterString
+ bronnauwkeurigheid: Real
+ brontype: Brontype
+ dimensie: Dimensie
+ identificatie: CharacterString
+ objectBeginTijd: DateTime
+ objectEindTijd: DateTime [0..1]
+ tdnCode: Integer [0..1]
+ versieBeginTijd: DateTime
+ versieEindTijd: DateTime [0..1]
+ visualisatieCode: Integer [0..1]
«FeatureType»
Wegdeel::Wegdeel
«FeatureType»
Spoorbaandeel::Spoorbaandeel
«FeatureType»
Waterdeel::Waterdeel
«FeatureType»
FunctioneelGebied::FunctioneelGebied
«FeatureType»
GeografischGebied::GeografischGebied
«FeatureType»
Terrein::Terrein
«FeatureType»
RegistratiefGebied::RegistratiefGebied
«FeatureType»
Gebouw::Gebouw
«FeatureType»
Relief::Relief
«FeatureType»
Inrichtingselement::Inrichtingselement
«FeatureType»
Relief::IsoHoogte
«FeatureType»
Relief::Hoogteverschil
«FeatureType»
Relief::HoogteOfDieptePunt
«FeatureType»
Relief::OverigRelief
«FeatureType»
Relief::KadeOfWal

C Data model comparison: attribute values

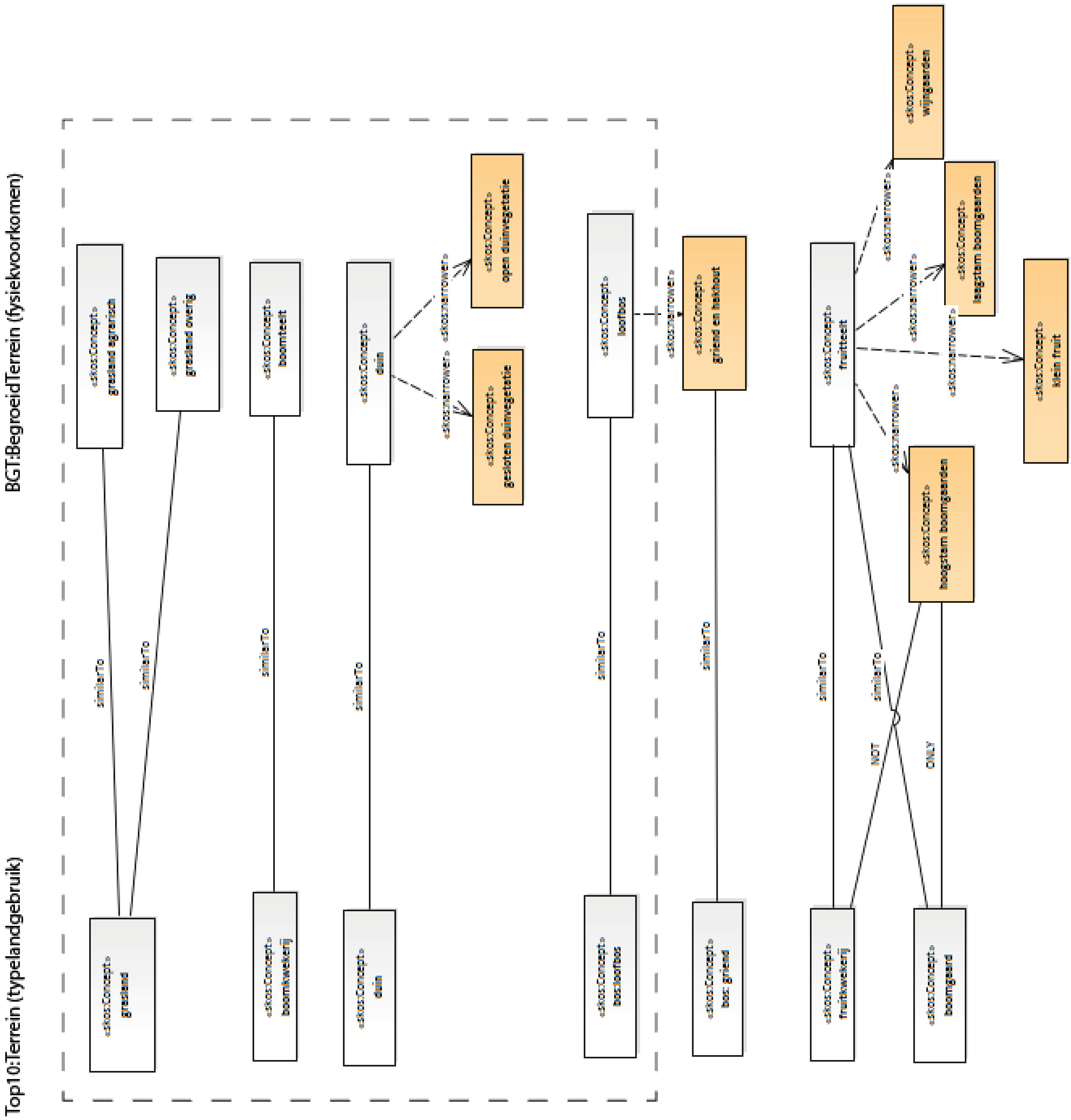

BGT:BegroeidTerrein (fysiekvoorkomen)
Top10:Terrein (typelandgebruik)
«skos:Concept» grasland agrarisch
«skos:Concept» grasland overig
«skos:Concept» boomteelt
«skos:Concept» duin
«skos:Concept» open duinvegetatie
«skos:Concept» gesloten duinvegetatie
«skos:Concept» loofbos
«skos:Concept» griend en hakhout
«skos:Concept» fruitteelt
«skos:Concept» wijngaarden
«skos:Concept» laagstam boomgaarden
«skos:Concept» klein fruit
«skos:Concept» hoogstam boomgaarden
«skos:narrower»
similarTo
NOT
ONLY
«skos:Concept» grasland
«skos:Concept» boomkwekerij
«skos:Concept» duin
«skos:Concept» bos: loofbos
«skos:Concept» bos: griend
«skos:Concept» fruitkwekerij
«skos:Concept» boomgaard

Figure C.1: Within the frame are possibly ’stable’ mappings, outside are ’unstable’ mappings: with top10 values possibly mapping to plus-type values.

D Top10NL tables

Class and attributes	values	B1	B2	B3	B4	B5	All sets
Gebouw							
type gebouw	Bezoekerscentrum	2.78	5.56	5.56	●	●	13.89
	Boortoren	●	●	●	●	18.18	18.18
	Brandtoren	●	●	●	●	●	●
	Brandweerkazerne	3.78	6.76	1.29	2.49	8.85	23.16
	Bunker	18.78	1.34	3.73	●	1.79	25.63
	Crematorium	1.82	7.27	●	3.64	9.09	21.82
	Dok	●	●	●	●	27.08	27.08
	Elektriciteitscentrale	3.03	●	●	●	3.03	6.06
	Fabriek	0.40	17.00	1.58	1.98	6.72	27.67
	Fort	9.84	●	●	1.64	●	11.48
	Gemaal	2.06	1.66	0.97	0.20	11.21	16.10
	Gemeentehuis	3.72	6.81	1.24	6.50	9.91	28.17
	Gevangenis	11.38	1.63	1.63	●	18.70	33.33
	Hotel	14.29	●	●	●	●	14.29
	Huizenblok	7.63	2.35	0.20	1.58	27.51	39.29
	Kapel	0.38	1.01	0.13	7.35	0.63	9.51
	Kasteel	4.47	9.50	●	3.91	1.68	19.55
	KasWarenhuis	1.16	3.57	0.23	1.84	27.03	33.84
	Kerk	4.71	6.54	1.16	2.08	12.37	26.87
	KerncentraleKernreactor	●	●	●	●	●	●
	KliniekInrichtingSanatorium	8.33	25.00	●	12.50	●	45.83
	Klokkentoren	5.26	5.26	●	5.26	15.79	31.58
	KloosterAbdij	2.74	4.11	●	4.11	●	10.96
	Koeltoren	●	●	●	●	9.09	9.09
	Koepel	57.14	●	●	●	●	57.14
	Kunstijsbaan	10.00	●	●	●	10.00	20.00
	Lichttoren	●	●	●	●	●	●
	Luchtwachttoren	●	●	●	●	●	●
	Manege	2.34	7.17	1.56	3.89	6.85	21.81
	MarkantGebouw	●	10.00	10.00	●	40.00	60.00
	MilitairGebouw	●	●	●	●	●	●
	Moskee	8.66	5.07	0.60	2.69	15.22	32.24
	Museum	2.33	6.98	2.91	2.33	6.40	20.93
	Observatorium	●	●	●	●	100.00	100.00
	Overig_gebouw	3.11	7.07	1.51	3.70	6.97	22.35
	OverigReligieusGebouw	11.90	4.76	1.19	2.38	15.48	35.71
	Paleis	50.00	50.00	●	●	●	100.00
	Parkeerdak/dek/garage	8.08	3.58	0.82	5.21	22.29	39.98
	Peilmeetstation	●	25.00	●	●	●	25.00
	Politiebureau	3.81	6.27	1.36	2.72	15.53	29.70
	Pompstation	2.76	5.52	2.07	2.76	8.97	22.07
	Postkantoor	3.10	5.48	1.27	3.26	8.98	22.08
	Psychiatrisch Ziekenhuis/Centrum	4.35	5.80	4.35	15.94	10.14	40.58
	Radarpost	2.94	●	●	●	28.43	31.37
	Radartoren	●	5.26	●	●	10.53	15.79
	RadiotorenTelevisietoren	6.67	●	6.67	6.67	6.67	26.67
	Recreatiecentrum	●	●	●	●	●	●

	Reddingboothuisje	●	●	●	●	●	●
	Remise	12.50	12.50	●	12.50	37.50	75.00
	Rune	●	1.79	●	1.79	7.14	10.71
	Schaapskooi	●	17.86	7.14	●	●	25.00
	School	7.26	6.15	2.79	●	10.06	26.26
	Schoorsteen	2.95	5.79	0.44	3.17	12.35	24.70
	Silo	●	5.26	●	●	5.26	10.53
	Sporthal	4.00	5.90	1.36	3.54	9.72	24.52
	Stadion	2.94	2.94	●	2.94	5.88	14.71
	StadskantoorHulpsecretarie	2.20	●	●	2.20	21.98	26.37
	Stationsgebouw	4.85	5.22	0.37	2.24	27.61	40.30
	Synagoge	●	7.14	●	●	10.71	17.86
	Tank	0.84	2.96	1.66	1.78	22.74	29.97
	Tankstation	3.87	6.50	1.35	3.61	10.96	26.30
	Telecommunicatietoren	●	25.00	●	●	25.00	50.00
	Tol_gebouw	●	●	●	●	●	●
	Toren	4.20	5.44	0.19	5.44	7.82	23.09
	Transformatorstation_gebouw	1.08	4.86	0.54	2.70	30.27	39.46
	Uitzichttoren	1.31	8.50	5.88	0.65	9.15	25.49
	Universiteit	●	●	●	15.38	35.38	50.77
	Veiling	●	●	●	●	39.13	39.13
	Verkeerstoren	●	7.69	●	7.69	15.38	30.77
	Vuurtoren	●	●	●	●	7.41	7.41
	Waterradmolen	●	7.32	●	12.20	●	19.51
	Watertoren	6.55	2.38	0.60	0.60	11.31	21.43
	Wegrestaurant	●	●	●	●	●	●
	Werf_gebouw	●	●	●	●	●	●
	Windmolen	1.89	3.77	1.89	1.89	11.32	20.75
	WindmolenKorenmolen	1.10	5.83	0.94	3.62	7.09	18.58
	WindmolenWatermolen	1.58	0.53	●	●	25.33	27.44
	Windturbine	●	0.94	0.05	●	1.87	2.86
	Zendtoren	●	●	●	●	25.00	25.00
	Ziekenhuis	6.50	2.00	2.00	4.50	22.50	37.50
	Zwembad_gebouw	4.12	7.61	2.06	4.75	10.14	28.68
fysiek voorkomen (O)	ondergronds	4.40	3.91	1.22	3.67	24.45	37.65
	overkluisd	14.29	7.14	7.14	●	7.14	35.71
hoogteklasse	laagbouw	3.10	6.98	1.50	3.65	7.28	22.52
	hoogbouw	6.24	4.03	0.53	3.88	32.83	47.51
gebruiksdoel (O)	bijeenkomstfunctie	●	●	6.25	●	25.00	31.25
	celfunctie	●	●	●	●	●	●
	gezondheidszorgfunctie	●	●	●	●	11.11	11.11
	industriefunctie	●	●	20.00	20.00	●	40.00
	kantoorfunctie	●	●	●	●	●	●
	logiesfunctie	●	●	●	●	●	●
	onderwijsfunctie	●	9.09	●	●	●	9.09
	sportfunctie	5.56				5.56	11.11
	winkelfunctie	●	●	●	●	●	●
	woonfunctie	●	●	●	●	50.00	50.00
	overige gebruiksfunctie	14.29	7.14		7.14	7.14	35.71
Functioneel gebied							
type functioneel gebied	Arboretum	●	●	●	●	●	●
	Attractiepark	●	3.92	1.96	1.96	9.80	17.65
	Bedrijventerrein	3.46	5.55	1.09	3.16	11.81	25.06
	Begraafplaats	2.39	4.20	1.15	2.82	5.79	16.34
	Boswachterij	●	●	36.36	●	●	36.36
	Botanische tuin	5.56	16.67	3.70	1.85	12.96	40.74
	Bungalowpark	3.11	16.48	2.64	0.47	3.30	25.99
	Camping, Kampeerterrein	1.14	4.84	3.56	1.66	1.28	12.48
	Campus	6.00	10.00	●	6.00	18.00	40.00

Caravanpark	3.21	18.18	●	0.53	8.02	29.95
Circuit	5.33	5.33	5.33	2.67	6.67	25.33
Crossbaan	2.35	6.47	1.76	5.29	4.12	20.00
Dierentuin, Safaripark	1.72	8.62	●	6.90	12.07	29.31
Eendenkooi	1.63	●	●	●	3.80	5.43
Emplacement	5.16	4.91	0.49	1.23	13.76	25.55
Erebegraafplaats	1.92	7.69	●	5.77	5.77	21.15
Gaswinning	0.24	0.73	3.17	●	2.68	6.83
Gebied voor radioastronomie	●	●	●	●	●	●
Gebied Met Hoge Objecten	●	●	●	●	●	●
Gebouwencomplex	4.86	6.25	3.47	10.42	13.19	38.19
Golfterrein	6.16	6.16	1.45	5.43	9.78	28.99
Grafheuvel	5.90	34.38	18.03	2.86	●	61.17
Grindwinning	●	10.00	●	●	●	10.00
Groeve	●	●	●	●	●	●
Haven	3.06	1.53	●	0.44	18.38	23.41
Heemtuin	3.55	8.51	0.71	5.67	14.89	33.33
Helikopterlandingsterrein	3.03	13.13	●	3.03	7.07	26.26
Ijsbaan	1.92	2.56	3.96	0.26	4.99	13.68
Infiltratiegebied	●	12.50	●	●	25.00	37.50
Jachthaven	5.14	2.01	0.16	0.08	10.37	17.77
Kartingbaan	6.67	●	●	6.67	●	13.33
Kassengebied	●	0.97	●	0.97	17.48	19.42
Kazerne, Legerplaats	18.75	14.58	2.08	2.08	8.33	45.83
Landgoed	2.56	13.68	4.27	1.71	3.42	25.64
Mijn	●	●	●	●	●	●
Milieustraat	3.95	5.26	0.79	3.95	10.79	24.74
Militair Oefengebied, Schietterrein	7.94	19.05	7.94	1.59	1.59	38.10
Mosselbank	●	●	●	●	●	●
Nationaal park	●	9.52	●	●	●	9.52
Natuurgebied	2.56	1.79	1.28	0.77	2.05	8.46
Natuurgebied, Natuurreservaat	●	●	●	●	●	●
Oliewinning	●	●	●	●	20.93	20.93
Openluchtmuseum	1.49	1.49	1.49	4.48	5.97	14.93
Openluchttheater	6.25	9.38	4.69	7.81	1.56	29.69
Overig	●	●	●	●	●	●
Park	5.07	5.55	0.64	3.70	13.35	28.32
Plantsoen	●	●	●	●	●	●
Productie-installatie	●	5.08	●	2.54	10.17	17.80
Recreatiegebied	2.73	2.55	1.46	1.09	9.65	17.49
Renbaan	●	●	●	●	10.00	10.00
Skibaan	5.71	5.71	●	5.71	8.57	25.71
Slipschool	●	●	●	7.14	14.29	21.43
Sluizencomplex	2.92	1.36	3.89	1.95	6.03	16.15
Sportterrein, Sportcomplex	3.45	4.68	1.20	3.23	10.34	22.90
Stortplaats	4.35	8.70	●	●	●	13.04
Tennispark	4.67	4.87	1.04	3.94	10.47	24.99
Transformatorstation	4.67	4.04	0.50	1.51	21.44	32.16
Tuincentrum	3.88	6.65	1.48	3.88	10.91	26.80
Vakantiepark	1.61	10.63	2.41	1.88	2.59	19.12
Verdedigingswerk	7.22	4.12	●	●	●	11.34
Verzorgingsplaats	1.28	8.70	2.05	2.81	7.16	21.99
Viskwekerij	●	20.00	●	10.00	●	30.00
Visvijvercomplex	●	3.64	3.64	7.27	●	14.55
Vliegveld, Luchthaven	3.85	7.69	●	3.85	3.85	19.23
Volkstuinen	3.59	5.23	0.90	2.33	11.19	23.23
Waterkering	●	●	●	●	15.79	15.79
Werf	●	1.23	●	●	16.05	17.28
Wildwissel	16.25	18.75	2.50	7.50	1.25	46.25

	Windturbinepark	●	1.07	●	●	2.85	3.91
	Woonwagencentrum	4.75	8.66	0.92	4.83	8.49	27.64
	Zandwinning	●	4.17	6.94	●	●	11.11
	Zenderpark	●	●	●	●	●	●
	Ziekenhuiscomplex	6.90	4.31	0.86	4.31	18.97	35.34
	Zonnepark	1.56	4.69	4.69	1.56	1.56	14.06
	Zoutwinning	●	●	●	●	●	●
	Zuiveringsinstallatie	3.54	5.09	1.33	1.55	7.96	19.47
	Zweefvliegveldterrein	6.67	10.00	●	3.33	3.33	23.33
	Zwembadcomplex	2.11	9.08	2.11	2.76	6.97	23.03
Terrein							
type landgebruik	Aanlegsteiger_terrein	1.68	1.21	0.09	0.56	23.64	27.20
	Populieren	1.03	2.32	0.45	5.02	6.47	15.30
	BosGriend	3.23	1.58	0.30	●	14.70	19.80
	Overig_terrein	3.32	6.30	1.48	3.45	9.40	23.94
	BosLoofbos	3.19	5.67	3.04	3.23	6.60	21.73
	Heide	7.02	21.13	11.14	7.08	0.03	46.40
	Boomgaard	2.28	7.53	0.51	1.02	2.02	13.36
	Akkerland	0.54	3.87	2.38	3.45	2.36	12.60
	Dodenakker	3.39	4.94	1.56	2.46	7.03	19.38
	BosNaaldbos	7.06	18.33	4.01	10.81	0.42	40.63
	Zand	3.58	13.61	1.90	3.14	3.11	25.34
	Spoorbaanlichaam	5.75	5.34	0.49	1.41	16.06	29.04
	BasaltblokkenSteenglooiing	0.33	7.77	●	0.12	11.06	19.28
	DodenakkerMetBos	3.07	57.67	0.61	1.84	1.23	64.42
	BebouwdGebied	7.48	2.42	0.23	1.59	27.61	39.34
	Grasland	2.66	4.77	1.91	2.77	7.24	19.36
	Braakliggend	1.24	2.07	1.24	2.89	4.55	11.98
	BosGemengdBos	6.36	16.62	3.83	7.07	0.28	34.16
	Duin	●	●	0.02	●	14.70	14.71
	Fruitkwekerij	1.45	4.39	0.19	0.98	1.84	8.86
	Boomkwekerij	1.27	8.04	0.98	5.62	7.36	23.27
fysiek voorkomen (O)	overkluisd	6.50	5.17	1.17	4.50	27.83	45.17
	in tunnel	9.42	7.33	●	3.14	25.13	45.03
	op vast deel van brug	4.13	2.00	0.35	1.71	17.27	25.46
	op beweegbaar deel van brug		2.13	1.42	1.42	20.57	25.53
voorkomen (O)	met riet	2.90	1.24	0.59	1.45	6.62	12.80
	dras, moerassig	1.91	3.01	8.96	2.92	2.38	19.18
Waterdeel							
type water	BronWel	●	●	●	●	●	●
	Droogvallend	●	●	●	●	1.33	1.33
	'Droogvallend (LAT)'	●	●	●	●	1.50	1.50
	GreppelDrogeSloot	0.89	6.26	4.93	5.71	0.51	18.31
	MeerPlas	2.52	5.09	2.48	3.26	5.67	19.02
	Overig_waterdeel	●	●	●	●	●	●
	Waterloop	2.48	2.54	1.41	1.26	7.87	15.56
	Zee_waterdeel	●	●	●	●	3.33	3.33
fysiek voorkomen (O)	in sluis	2.34	1.11	2.12	1.11	8.36	15.05
	op brug	●	12.50	●	●	20.83	33.33
	in duiker	1.40	4.39	2.72	1.93	6.02	16.45
	in afsluitbare duiker	1.58	1.83	1.39	0.68	6.66	12.14
	in grondduiker	0.72	2.15	2.86	2.74	6.91	15.38
	in afsluitbare grondduiker	0.53	3.16	6.32	1.58	5.79	17.37
	overkluisd	6.12	1.05	0.31	1.41	13.60	22.49
functie	drinkwaterbekken	●	●	●	●	10.34	10.34
	haven	1.93	2.04	0.11	0.11	19.43	23.62
	natuurbad	4.65	2.33	●	4.65	4.65	16.28
	viskwekerij	●	12.50	●	29.69	●	42.19
	vistrap	●	3.78	2.43	1.08	0.81	8.11

	vloeiveld	●	●	2.22	●	●	2.22
	waterval	●	100.00	●	●	●	100.00
	waterzuivering	3.28	7.70	1.22	2.78	7.11	22.08
	zwembad	1.63	8.00	1.86	2.48	5.75	19.72
	overig	1.98	3.81	2.57	2.74	5.45	16.55
	onbekend	●	●	●	●	●	●

E BGT tables

Attributes	values	B1	B2	B3	B4	B5	All sets
Gebouw (Pand)							
identificatiebagpnd	(amount of pand objects)						
Functioneel gebied							
bgt type	kering	0.92	0.68	0.24		13.58	15.42
	niet-bgt	1.70	10.97	1.40	10.45	5.36	29.88
plus type (O)	bedrijvigheid	●	0.33	●	●	●	0.33
	begraafplaats	0.81	24.39	0.63	1.35	2.97	30.15
	verzorgingsplaats	●	●	●	●	●	●
	bewoning	●	●	●	●	●	●
	recreatie: park	●	●	●	●	●	●
	recreatie: bungalowpark	●	42.86	●	●	●	42.86
	maatschappelijke/publieksvoorziening	●	●	●	●	0.78	0.78
	functioneel beheer	0.03	●	4.59	7.09	14.26	25.97
	bushalte	0.74	0.83	1.02	9.05	5.72	17.36
	infrastructuur verkeer en vervoer	●	●	●	●	85.33	85.33
	carpoolplaats	●	25.00	●	●	●	25.00
	recreatie: volkstuin	●	9.76	0.81	0.81	4.07	15.45
	natuur en landschap	●	●	25.00	●	●	25.00
	waterbergingsgebied	●	14.00	●	71.00	2.00	87.00
	recreatie: camping	●	11.29	3.23	●	●	14.52
	infrastructuur waterstaatswerken	●	●	●	●	●	●
	hondenuitlaatplaats	●	28.36	●	44.73	0.08	73.17
	recreatie: speeltuin	5.10	15.58	●	5.52	0.28	26.48
	landbouw	●	●	●	●	●	●
	recreatie: sportterrein	2.83	9.79	●	6.56	2.93	22.10
	benzinestation	●	33.33	●	●	●	33.33
	recreatie	0.91	●	1.52	●	●	2.44
Terrein (Begroeid Terreindeel)							
bgt fysiekvoorkomen	transitie	●	0.81	●	●	0.04	0.84
	bouwland	0.53	3.72	2.23	3.81	3.86	14.15
	rietland	0.83	0.91	0.29	0.65	6.67	9.34
	boomteelt	0.73	9.20	0.37	5.42	18.56	34.28
	duin	2.98	●	●	●	27.79	30.77
	fruitteelt	1.46	5.45	0.18	1.27	1.71	10.07
	naaldbos	6.39	34.44	4.52	10.49	0.08	55.92
	groenvoorziening	5.45	5.99	1.22	4.55	13.01	30.23
	grasland overig	1.68	6.28	1.87	1.92	5.86	17.60
	houtwal	0.63	4.02	2.13	1.03	3.34	11.16
	kwelder	●	0.03	0.11	0.19	0.92	1.24
	moeras	0.75	3.60	1.50	2.32	3.63	11.80
	gemengd bos	5.25	15.39	3.04	6.15	1.91	31.73
	grasland agrarisch	2.08	4.35	1.76	2.53	3.91	14.63
	struiken	2.23	3.14	0.43	8.56	18.68	33.04
	loofbos	3.39	8.61	5.39	3.74	4.26	25.40
	heide	8.40	29.67	8.48	4.94	0.07	51.57
plus fysiekvoorkomen (O)	akkerbouw	0.09	7.64	0.73	6.91	3.75	19.11
	bollenteelt	●	7.69	11.54	7.69	23.08	50.00
	braakliggend	●	6.42	0.72	11.15	27.83	46.11
	vollegrondsteelt	0.38	3.01	0.06	0.64	36.07	40.17

	gesloten duinvegetatie	●	●	●	●	29.41	29.41
	open duinvegetatie	16.35	●	●	●	20.79	37.15
	hoogstam boomgaarden	0.43	18.18	0.43	1.73	●	20.78
	wijngaarden	●	●	●	50.00	●	50.00
	klein fruit	●	4.65	●	6.98	●	11.63
	laagstam boomgaarden	2.16	12.23	0.72	13.67	2.88	31.65
	bodembedekkers	3.70	8.06	0.44	6.94	10.92	30.05
	struikrozen	1.46	3.45	0.50	4.47	15.50	25.39
	bosplantsoen	3.04	6.69	1.94	3.69	12.73	28.09
	planten	1.92	4.41	1.59	0.72	24.74	33.39
	gras- en kruidachtigen	3.38	5.96	1.84	4.31	13.42	28.91
	heesters	2.74	5.96	1.19	5.34	14.89	30.13
	griend en hakhout	3.77	4.31	0.36	0.39	39.77	48.60
Terrein (Onegroeid Terreindeel)							
bgt fysiekvoorkomen	erf	5.04	3.23	1.06	2.53	24.50	36.36
	half verhard	3.63	4.93	1.27	1.69	10.67	22.18
	open verharding	4.25	8.00	0.71	1.56	12.63	27.15
	onverhard	2.46	6.59	0.97	3.17	21.14	34.32
	transitie	●	23.49	0.67	●	1.01	25.17
	gesloten verharding	4.53	2.88	0.65	2.17	21.97	32.20
	zand	6.11	4.65	0.65	6.30	11.82	29.53
plus fysiekvoorkomen (O)	asfalt	3.33	7.99	0.75	1.60	13.15	26.82
	cementbeton	1.94	2.15	1.70	1.33	27.62	34.74
	kunststof	3.88	2.24	0.20	2.04	32.56	40.91
	grasklinkers	4.73	6.73	1.06	0.81	11.33	24.65
	gravel	2.68	5.08	0.90	4.01	13.86	26.52
	grind	2.96	6.45	0.75	1.52	14.56	26.23
	puin	0.82	17.20	0.30	0.60	7.35	26.26
	schelpen	1.92	6.15	0.82	1.28	9.25	19.42
	beton element	2.76	6.51	1.20	2.15	11.11	23.73
	betonstraatstenen	5.09	2.98	1.09	2.20	20.70	32.05
	gebakken klinkers	3.13	2.59	0.94	3.86	9.76	20.27
	sierbestrating	4.55	5.15	1.48	2.61	16.97	30.75
	tegels	0.38	1.27	0.25	1.39	1.77	5.06
	strand en strandwal	0.78	8.20	●	0.13	66.80	75.91
	zandverstuiving	3.10	9.07	4.42	0.13	13.27	29.99
	boomschors	1.77	11.08	0.70	5.06	13.77	32.39
	zand	0.19	0.24	2.94	0.05	13.71	17.13
Water (Waterdeel)							
bgt type	greppel, droge sloot	0.49	3.27	2.83	3.85	0.80	11.23
	transitie		1.31				1.31
	waterloop	1.69	4.00	2.49	2.80	7.33	18.31
	watervlakte	1.79	6.04	2.64	4.32	2.64	17.43
	zee				0.11	1.26	1.37
plus type (O)	beek	0.05	24.59	●	12.86	20.78	58.29
	bron	●	8.57	●	●	●	8.57
	gracht	0.25	2.99	●	0.75	3.31	7.30
	kanaal	0.45	0.49	8.93	0.55	6.66	17.07
	rivier	9.99	0.71	0.56	4.60	27.28	43.14
	sloot	1.71	6.59	4.27	3.69	5.59	21.85
	haven	0.94	1.72	0.31	●	15.13	18.10
	meer, plas, ven, vijver	1.99	9.00	4.01	4.85	2.43	22.27
Water (Ondersteunend waterdeel)							
bgt type	oever, slootkant	1.92	2.36	1.90	2.37	15.57	24.13
	slik	0.05	0.16	0.11		2.75	3.07
	transitie	●	●	●	●	●	0.00

F Syntax matching

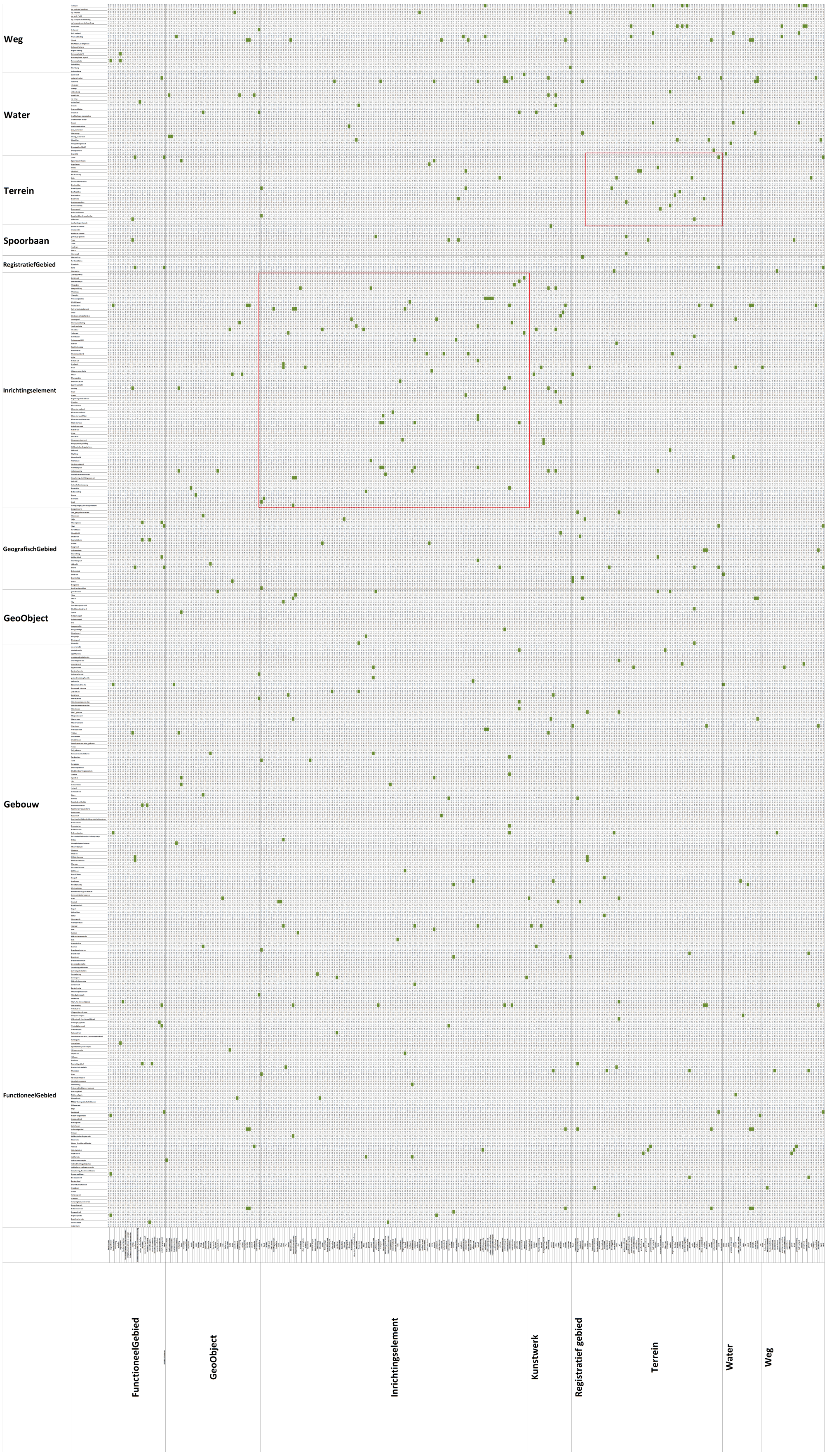


Figure F.1: The vertical axis represents TOP10NL concepts and the horizontal axis represents BGT concepts.

G SPARQL CONSTRUCT - example for retrieving intersecting features

```
PREFIX top10: <http://brt.basisregistraties.overheid.nl/def/top10nl#>
PREFIX bgt: <http://definities.geostandaarden.nl/def/imgeo#>
PREFIX maps: <http://examples.com/overlap_mappings#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX gsp: <http://www.opengis.net/ont/geosparql#>
PREFIX geof: <http://www.opengis.net/def/function/geosparql/>
PREFIX f: <java:com.thesis.functions.>

CONSTRUCT {
maps:relation1 rdf:type maps:OverlapRelations ;
maps:mainfeature ?top10_obj ;
maps:maintype ?q ;
maps:hasOverlap _:blank .
_:blank maps:secfeature ?bgtz ;
maps:mainRelativeArea ?top10_rel ;
maps:secRelativeArea ?bgt_rel ;
a ?p .
} WHERE {
VALUES ?top10_obj {%s}
FILTER (geof:sfIntersects(?topgeom, ?bgtgeom) = true).
SERVICE <https://data.pdok.nl/sparql>
{?top10_obj top10:geometrieVlak | top10:geometrie ?geomz .
?geomz gsp:asWKT ?topgeom .}
?bgtz rdf:type %s ;
gsp:hasGeometry ?geomz2 ;
foaf:isPrimaryTopicOf ?objectregistratie .
FILTER NOT EXISTS { ?objectregistratie bgt:eindRegistratie ?value }
?geomz2 gsp:asWKT ?bgtgeom .

BIND (f:GeoFunctions(?topgeom, ?bgtgeom) AS ?top10_rel)
BIND (f:GeoFunctions(?bgtgeom, ?topgeom) AS ?bgt_rel)
}
```

H SHACL/SPARQL CONSTRUCT defining topological relations

```
ex:OverlapRelations
        rdf:type rdfs:Class , sh:NodeShape ;
        sh:targetClass maps:OverlapRelations;
        rdfs:label "whether overlap between features is enough to generate relation" ;
        sh:rule [
                rdf:type sh:SPARQLRule ;
                sh:prefixes [sh:declare [
                sh:prefix "maps" ;
                sh:namespace "http://examples.com/overlap_mappings#"^^xsd:anyURI ;
        ] ],
        [ sh:declare [
                sh:prefix "align" ;
                sh:namespace "http://alignment_classes.com/#"^^xsd:anyURI ;
        ] ];
                sh:construct """
                PREFIX maps: <http://examples.com/overlap_mappings#>
                PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
                PREFIX align: <http://alignment_classes.com/#>
                PREFIX rel: <http://example.com/relations#>
                PREFIX owl:    <http://www.w3.org/2002/07/owl#>
                PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
                PREFIX dc: <http://purl.org/dc/terms/>
                CONSTRUCT {
                ?top10obj  rdf:type ?top10class ;
                ?align_rel ?bgt_obj ;
                align:has ?bgtclass .

                }
                WHERE {
                $this maps:maintype ?top10class ;
                          maps:mainfeature ?top10obj .

                ?x maps:secfeature ?bgt_obj ;
                        rdf:type ?bgtclass ;
                        maps:mainRelativeArea ?mainarea ;
                        maps:secRelativeArea ?secarea .

                        {SELECT (COUNT(?bgt_obj) AS ?count)
                        (SUM(?secarea) AS ?sum_sec)(SUM(?mainarea) AS ?sum_main)
                                WHERE {

                                                ?x maps:secfeature ?bgt_obj ;
                                                rdf:type ?bgtclass ;
                                                maps:mainRelativeArea ?mainarea ;
                                                maps:secRelativeArea ?secarea .
                                                FILTER((?mainarea >= 80 && ?secarea >= 80)
                                                                ||(?mainarea < 80 && ?secarea >= 80)
                                                                ||(?mainarea >= 80 && ?secarea < 80)) .
                                                }
                                }
                BIND((?sum_sec/?count) as ?blue).
                OPTIONAL {
                FILTER(?top10_alignedclass = ?top10class && ?bgt_alignedclass = ?bgtclass)}

                BIND (IF((?count = 1 && ?mainarea >= 80 && ?secarea >= 80), align:equals,
IF((?count = 1 && ?mainarea >= 80 && ?secarea < 80), align:is_contained,
IF((?count = 1 && ?mainarea < 80 && ?secarea >= 80), align:contains,
IF((?count > 1 && (?sum_main > 80 || ?blue > 70)), align:aggregates,
" "
)))) AS ?align_rel)

        }
""";

        ].
```

I SHACL/SPARQL CONSTRUCT example of constraint based on alignment

```
top10:_Top10NLObject
        rdf:type owl:Class, sh:NodeShape ;
        sh:severity sh:Warning ;
        sh:sparql [
                rdf:type sh:SPARQLConstraint ;
                sh:message "Top10 object {?top10class} and BGT object {?bgtclass}
                are never found together" ;
                sh:select """
                PREFIX rdf: <http://www.w3.org/1999/02/22−rdf−syntax−ns#>
                PREFIX align: <http://alignment_classes.com/#>
                PREFIX rel: <http://example.com/relations#>
                PREFIX dc: <http://purl.org/dc/terms/>

                SELECT DISTINCT $this ?top10class ?bgtclass
                WHERE {
                $this rdf:type ?top10class ;
                          align:has ?bgtclass .

                ?alignrule2 align:entity1 ?top10class ;
                        align:entity2 ?bgtclass ;
                        align:relation rel:sem_disjoint ;
                        dc:subject ?topo .

                        }
                """;

                        ].
```

Bibliography

Alani, H., Dupplaw, D., Sheridan, J., O'Hara, K., Darlington, J., Shadbolt, N., and Tullo, C. (2007). Unlocking the potential of public sector information with semantic web technology. In *The Semantic Web: 6th International Semantic Web Conference, 2nd Asian Semantic Web Conference*. Springer.

Baglatzi, A. and Kuhn, W. (2013). On the formulation of conceptual spaces for land cover classification systems. In *Geographic Information Science at the Heart of Europe*, pages 173–188. Springer International Publishing.

Ballatore, A., Bertolotto, M., and Wilson, D. (2014). An evaluative baseline for geo-semantic relatedness and similarity. *GeoInformatica*, 18.

Berners-Lee, T. (2019). Linked data - design issues. `https://www.w3.org/DesignIssues/LinkedData.html`. Accessed 10 Nov 2019.

Berners-Lee, T., Hendler, J., and Lassila, O. (2001). The semantic web: A new form of web content that is meaningful to computers will unleash a revolution of new possibilities. *ScientificAmerican.com*.

Brink, L. v. d. (2018). *Geospatial Data on the Web*. PhD thesis, Delft University of Technology.

Car, N. J., Box, P. J., and Sommer, A. (2019). The location index: A semantic web spatial data infrastructure. In Hitzler, P., Fernández, M., Janowicz, K., Zaveri, A., Gray, A. J., Lopez, V., Haller, A., and Hammar, K., editors, *The Semantic Web*, pages 543–557, Cham. Springer International Publishing.

Čerba, O. and Jedlička, K. (2016). Linked forests: Semantic similarity of geographical concepts "forest". *Open Geosciences*, 8(1).

Chen, Y., Sabri, S., Rajabifard, A., and Agunbiade, M. E. (2018). An ontology-based spatial data harmonisation for urban analytics. *Computers, Environment and Urban Systems*, 72:177–190.

Cohen, W. W., Ravikumar, P., and Fienberg, S. E. (2003). A comparison of string distance metrics for name-matching tasks. In *Proceedings of the 2003 International Conference on Information Integration on the Web*, IIWEB'03, page 73–78. AAAI Press.

Eiter, T., Ianni, G., Polleres, A., Schindlauer, R., and Tompits, H. (2006). *Reasoning with Rules and Ontologies*, pages 93–127. Springer Berlin Heidelberg, Berlin, Heidelberg.

Ekaputra, F., Sabou, M., Serral, E., Kiesling, E., and Biffl, S. (2017a). Ontology-based data integration in multi-disciplinary engineering environments: A review. *Open Journal of Information Systems (OJIS)*, 4:1–26.

Ekaputra, F., Sabou, M., Serral, E., Kiesling, E., and Biffl, S. (2017b). Ontology-based data integration in multi-disciplinary engineering environments: A review. *Open Journal of Information Systems (OJIS)*, 4:1–26.

en Landschap, W. N. (2019). Rapport werkgroep natuur en landschap. Technical report, Ministerie van Binnenlandse Zaken en Koninkrijksrelatie.

European Commission (2017). New european interoperability framework: Promoting seamless services and data flows for european public administrations. Technical report, European Commission.

European Commission (2020). Inspire knowledge base: Data specifications.

European Spatial Data Research (2018). Annual report. `http://www.eurosdr.net/sites/default/files/images/inline/eurosdr_annual_report_2018.pdf`. Accessed on 7 Nov 2019.

Euzenat (2018). Edoal: Expressive and declarative ontology alignment language. http://alignapi.gforge.inria.fr/edoal.html.

Euzenat, J. and Shvaiko, P. (2013). *Ontology Matching*. Springer Berlin Heidelberg.

Fonseca, F., Egenhofer, M., Davis, C., and Câmara, G. (2002). Semantic granularity in ontology-driven geographic information systems. *Annals of Mathematics and Artificial Intelligence*, 36(1/2):121–151.

Futia, G., Melandri, A., Vetrò, A., Morando, F., and Martin, J. C. D. (2017). Removing barriers to transparency: A case study on the use of semantic technologies to tackle procurement data inconsistency. In *The Semantic Web*, pages 623–637. Springer International Publishing.

Geonovum (2020). Nen3610 - linked data. https://geonovum.github.io/NEN3610-Linkeddata/.

Geonovum (n.d.). Nen 3610 basismodel voor informatiemodellen. https://www.geonovum.nl/geo-standaarden/nen-3610-basismodel-voor-informatiemodellen.

Halpin, H., Hayes, P. J., McCusker, J. P., McGuinness, D. L., and Thompson, H. S. (2010). When owl:sameAs isn't the same: An analysis of identity in linked data. In *Lecture Notes in Computer Science*, pages 305–320. Springer Berlin Heidelberg.

Hecht, B., Carton, S. H., Quaderi, M., Schöning, J., Raubal, M., Gergle, D., and Downey, D. (2012). Explanatory semantic relatedness and explicit spatialization for exploratory search. In *Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '12, page 415–424, New York, NY, USA. Association for Computing Machinery.

Homburg, T. and Boochs, F. (2019). Situation-dependent data quality analysis for geospatial data using semantic technologies. In *Business Information Systems Workshops*, pages 566–578. Springer International Publishing.

Huang, W., Raza, S. A., Mirzov, O., and Harrie, L. (2019). Assessment and benchmarking of spatially enabled RDF stores for the next generation of spatial data infrastructure. *ISPRS International Journal of Geo-Information*, 8(7):310.

Janowicz, K. (2012). Observation-driven geo-ontology engineering. *Transactions in GIS*, 16(3):351–374.

Janowicz, K., Simon, S., Pehle, T., and Hart, G. (2012). Geospatial semantics and linked spatiotemporal data –past, present, and future. *Semantic Web*, 3:321–332.

Jiang, J. J. and Conrath, D. W. (1997). Semantic similarity based on corpus statistics and lexical taxonomy. In *Proceedings of the 10th Research on Computational Linguistics International Conference*, pages 19–33, Taipei, Taiwan. The Association for Computational Linguistics and Chinese Language Processing (ACLCLP).

Kadaster (2020a). Basisregistratie topografie: Catalogus en productspecificaties.

Kadaster (2020b). Basisregistratie topografie: Catalogus en productspecificaties. https://zakelijk.kadaster.nl/documents/20838/88032/BRT+catalogus+productspecificaties/cb869308-5867-5a9d-626d-2fe290c7e4a6.

Kokla, M. and Guilbert, E. (2020). A review of geospatial semantic information modeling and elicitation approaches. *ISPRS International Journal of Geo-Information*, 9(3):146.

Kokla, M., Papadias, V., and Tomai, E. (2018). ENRICHMENT AND POPULATION OF a GEOSPATIAL ONTOLOGY FOR SEMANTIC INFORMATION EXTRACTION. *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XLII-4:309–314.

Koubarakis, M. and Kyzirakos, K. (2010). Modeling and querying metadata in the semantic sensor web: The model strdf and the query language stsparql. In *Proceedings of the 7th International Conference on The Semantic Web: Research and Applications - Volume Part I*, ESWC'10, page 425–439, Berlin, Heidelberg. Springer-Verlag.

Kovalenko, O. and Euzenat, J. (2016). Semantic matching of engineering data structures. In *Semantic Web Technologies for Intelligent Engineering Applications*, pages 137–157. Springer International Publishing.

Krötzsch, M., Simančík, F., and Horrocks, I. (2014). A description logic primer. In Lehmann, J. and Völker, J., editors, *Perspectives on Ontology Learning*. IOS Press.

Kuhn, W. (2003). Semantic reference systems. *International Journal of Geographical Information Science*, 17(5):405–409.

Kuhn, W. (2010). Modeling vs encoding for the semantic web. *Semantic Web*, 1(1,2):11–15.

Ministerie van Infrastructuur en Milieu (2013). Basisregistratie grootschalige topografie gegevenscatalogus. https://www.geonovum.nl/uploads/standards/downloads/BGTGegevenscatalogus111.pdf.

OGC (2011). Geosparql - a geographic query language for rdf data. https://www.ogc.org/standards/geosparql.

Parundekar, R., Knoblock, C., and Ambite, J. L. (2010). Aligning ontologies of geospatial linked data.

Rada, R., Mili, H., Bicknell, E., and Blettner, M. (1989). Development and application of a metric on semantic nets. *IEEE Transactions on Systems, Man, and Cybernetics*, 19(1):17–30.

Rajabifard, A., Feeney, M.-E. F., and Williamson, I. P. (2002). Future directions for SDI development. *International Journal of Applied Earth Observation and Geoinformation*, 4(1):11–22.

Resnik, P. (1999). Semantic similarity in a taxonomy: An information-based measure and its application to problems of ambiguity in natural language. *J. Artif. Int. Res.*, 11(1):95–130.

Richardson, R. and Smeaton, A. F. (1995). Using wordnet in a knowledge-based approach to information retrieval.

Rodríguez, M. A. and Egenhofer, M. J. (2004). Comparing geospatial entity classes: an asymmetric and context-dependent similarity measure. *International Journal of Geographical Information Science*, 18(3):229–256.

Ronzhin, Folmer, Maria, Brattinga, Beek, Lemmens, and van't Veer (2019). Kadaster knowledge graph: Beyond the fifth star of open data. *Information*, 10(10):310.

Santema and Brattinga (2018). Best practices for meaningful connected computing. https://bp4mc2.org/20181107/.

Sowa, J. F. (2000). Ontology, metadata, and semiotics. In Mineau, B. G. . G., editor, *Conceptual Structures: Logical, Linguistic, and Computational Issues, Lecture Notes in AI #1867*, pages 55–81, Berlin. ICCS'2000, Springer-Verlag. http://www.jfsowa.com/ontology/ontometa.htm.

Stoter, J. (2010). Afstemming imgeo-top10nl - basis voor een toekomstige multi-schaal topografische informatievoorziening. Technical report, Geonovum.

Sun, K., Zhu, Y., Pan, P., Hou, Z., Wang, D., Li, W., and Song, J. (2019). Geospatial data ontology: the semantic foundation of geospatial data integration and sharing. *Big Earth Data*, 3(3):269–296.

Sunna, W. and Cruz, I. F. (2007). Structure-based methods to enhance geospatial ontology alignment. In *GeoSpatial Semantics*, pages 82–97. Springer Berlin Heidelberg.

The Linked Open Data Cloud (2019). `https://lod-cloud.net/`. Accessed 10 Dec 2019.

Uschold, M. (2000). Creating, integrating and maintaining local and global ontologies. In *Proceedings of the First Workshop on Ontology Learning (OL-2000) in conjunction with the 14th European Conference on Artificial Intelligence (ECAI-2000)*. Citeseer.

Uschold, M. and Gruninger, M. (1996). Ontologies: Principles, methods and applications. *KNOWLEDGE ENGINEERING REVIEW*, 11:93–136.

van Infrastructuur en Milieu, M. (2013). Basisregistratie grootschalige topogra0e: Gegevenscatalogus bgt 1.0 februari 2012 basisregistratie grootschalige topografie gegevenscatalogus bgt 1.1.1.

van Loenen, B. (2018). Towards a user-oriented open data strategy. In *Open Data Exposed*, pages 33–53. T.M.C. Asser Press.

Vancauwenberghe, G. and van Loenen, B. (2017). Exploring the emergence of open spatial data infrastructures: Analysis of recent developments and trends in europe. In *Integrated Series in Information Systems*, pages 23–45. Springer International Publishing.

W3C (2007). Semantic web: Linked data on the web. https://www.w3.org/2007/Talks/0130-sb-W3CTechSemWeb/#(24).

Wache, H., Vögele, T., Visser, U., Stuckenschmidt, H., Schuster, G., Neumann, H., and Hübner, S. (2001). Ontology-based integration of information — a survey of existing approaches.

Wang, C., Ma, X., and Chen, J. (2018). Ontology-driven data integration and visualization for exploring regional geologic time and paleontological information. *Computers & Geosciences*, 115:12–19.

Worboys and Clementini (2001). Integration of imperfect spatial information. *Journal of Visual Languages & Computing*, 12(1):61–80.

Wu, Z. and Palmer, M. (1994). Verbs semantics and lexical selection. In *Proceedings of the 32nd annual meeting on Association for Computational Linguistics -*. Association for Computational Linguistics.

Yu, F., McMeekin, D. A., Arnold, L., and West, G. (2017). Semantic web technologies automate geospatial data conflation: Conflating points of interest data for emergency response services. In *Lecture Notes in Geoinformation and Cartography*, pages 111–131. Springer International Publishing.

Zhang, C., Zhao, T., and Li, W. (2010). Automatic search of geospatial features for disaster and emergency management. *International Journal of Applied Earth Observation and Geoinformation*, 12(6):409–418.

Colophon

This document was typeset using LaTeX, using the KOMA-Script class `scrbook`. The main font is Palatino.