D16.1: First Report on Data Mining

Author:

Wilcke,W.X. VU University Amsterdam

Ariadne is funded by the European Commission's
7th Framework Programme.

The research leading to these results has received funding from the European Community's Seventh Framework Programme (FP7-INFRASTRUCTURES-2012-1) under grant agreement n° 313193.'

Version: 1.2 *(final)* **March 2015**

Author: **Wilcke, W.X, VU University Amsterdam**

Contributing partners: **De Boer,V. -VU University Amsterdam**

Van Harmelen, F.A.H. -VU University Amsterdam

De Kleijn, M.T.M. -VU University Amsterdam

Wansleeben, M.- Leiden University

Quality Control Review: **Wright, H. E. – Archaeology Data Service, University of York**

ARIADNE is a project funded by the European Commission under the Community's Seventh Framework Programme, contract no. FP7

Table of Contents

Document History I

List of Abbreviations II

Executive Summary IV

Recommendations V

Roadmap VI

1 Introduction and Objectives 1

1.1 Structure of Report 1

2 Introduction to Linked Data 2

2.1 The RDF Data Model 2

2.2 Ontologies 5

2.3 The Semantic Web 6

2.4 Linked Archaeological Data 8

3 Introduction to Data Mining 10

3.1 Learning from Archaeological Data 10

3.2 Knowledge Discovery and Data Mining 12

3.3 Data Mining Tasks 13

3.4 Towards Mining the Semantic Web 15

4 Semantic-Web Mining 17

4.1 Data-Mining Tasks 17

4.2 Applicable Solutions 22

5 Domain Understanding 35

5.1 Relevant Studies 36

5.2 Wishes of Domain Experts 38

5.3 Summary 39

6 Data Understanding 40

6.1 Data Produced using Natural-Language Processing 40

6.2 Case Study on Data Repositories 41

6.3 Summary 43

7 Data Mining on Linked Archaeological Data 45

7.1 Hypothesis Generation 45

7.2 Assisted Query Formulation 46

7.3 Ranking of Query Results 46

7.4 Resource Recommender System 47

7.5 Data Quality Analysis 48

7.6 Trust Analysis 49

8 Conclusions 51

8.1 Domain Understanding 51

8.2 Data Understanding 52

8.3 Recommendations 53

8.4 Roadmap 54

Bibliography 55

Appendix A Reasoning with Logic i

A.1 Reasoning by Deduction i

A.2 Reasoning by Induction ii

A.3 Logic Reasoning within the Semantic Web ii

Appendix B Vector Space Models iv

Appendix C Learning Methods for Semantic Web Mining v

C.1 Propositional Learning v

C.2 Statistical Relational Learning vii

C.3 Kernel Methods xi

Appendix D Sample of Archaeological Scenarios xiv

Document History

1. 13th February, 2015 – Full Draft Version 1.0
2. 19th February, 2015 – QC Review 1.0
3. 4rd March, 2015 – Full Draft Version 1.1
4. 6th March, 2015 – QC Review 1.1
5. 6rd March, 2015 – Full Final Version 1.2

List of Abbreviations

The following abbreviations will be used in this report.

Abbreviation	*Full Term*
ACDM	ARIADNE Catalogue Data Model
ADS	Archaeological Data Service
API	Application Programming Interface
ARIADNE	Advanced Research Infrastructure for Archaeological Dataset Networking in Europe
BGV	Basic Geo Vocabulary
CAA	Computer Applications & Quantitative Methods in Archaeology
DCAT	Data Catalogue Vocabulary
DINAA	Digital Index of North-American Archaeology
DM	Data Mining
GIS	Geographic Information System
IG	Information Gain
ILP	Inductive Logic Programming
IT	Information Task
KDD	Knowledge Discovery and Data Mining
LAD	Linked Archaeological Data
LD	Linked Data
LOD	Linked Open Data
ML	Machine Learning
MRDM	Multi-Relational Data Mining
NLP	Natural Language Processing
OGC	Open Geospatial Consortium

Abbreviation	***Full Term***
OLAP	Online Analytical Processing
OWL	Web Ontology Language
PCA	Principal Component Analysis
PSL	Probabilistic Soft Logic
RDF	Resource Description Framework
RDFS	Resource Description Framework Schema
SKOS	Simple Knowledge Organization System
SRL	Statistical Relational Learning
SVM	Support Vector Machine
SW	Semantic Web
SWM	Semantic Web Mining
TL	Trust Level
UI	User Interface
URI	Universal Resource Indicator
VSM	Vector Space Models
W3C	World Wide Web Consortium
WGS	World Geodetic System
WP	Work Package

Executive Summary

ARIADNE, the Advanced Research Infrastructure for Archaeological Dataset Networking in Europe, will facilitate a central web portal that provides access to archaeological data from various sources in a standardized and open format. This format will likely adhere to the Linked Data paradigm either fully or partially, with the former being the option that we believe is needed to propel ARIADNE towards a higher level of interoperability. By this assumption, users will be able to use the portal to browse and search the data, thereby making use of all the advantages that Linked Data has to offer. Among these advantages are advanced search abilities, the inherent capability of drawing inferences, and the enrichment of data by linkage to and from external sources. We expect these features to have a positive impact on the archaeological research community. However, this does not necessarily add to the knowledge contained within the aggregated data sets. Ideally, we would like to expand this knowledge as well. One field of expertise that specializes in exactly that is data mining. Hereto, this field provides tools and techniques to identify *valid, novel, potentially useful, and ultimately understandable patterns in data* (U. M. Fayyad 1996).

This report examines the applicability and feasibility of integrating data mining solutions into ARIADNE. To this end, we explored various state-of-the-art theories, methods, and solutions to detect patterns in, and establish relations between, data from the archaeological domain. Throughout this report, we made the assumption that this data will adhere, either fully or partially, to the principles of the Linked Data paradigm. The subfield of data mining dedicated to this form of data, known as semantic web mining, was deliberately chosen over the-more-traditional tabular data mining, for its ability to fully exploit the graph-like structure of Linked Data without the loss of knowledge. In addition to data mining, our study focussed on usage-pattern analysis and content linking, as well as on information retrieval. To this end, a thorough analysis of users' needs and wishes was conducted, as well as an exploration of the expected data's characteristics. Furthermore, recent and relevant literature and experience on the topics involved were examined in depth.

The user-requirements study involved an analysis of the questionnaires and interviews that were conducted by work package 2.1 and 13.1, respectively. While providing valuable insight into the stakeholders of ARIADNE, both work packages only touched on the possibility of data mining. As a result, very little could be ascertained as to what path any data mining solution should follow. Moreover, the large majority of the stakeholders had little to no experience with data mining and were unaware of what it actually entailed. To mitigate this lack of direction, several additional interview sessions with stakeholders were held, during which the possibility of data mining was more-actively explored. Regardless, of all the topics discussed, only few were relevant with respect to data mining.

In its entirety, the requirements study seemed to indicate that the large majority of the difficulties experienced by stakeholders could be mitigated by the use of Linked Data alone. Several of these issues could additionally be improved upon even more with the help of data mining. These issues involved knowing which data is available, how to locate relevant data, and how to distil the relevant results from

those that are not. In addition, the quality of the data was mentioned prominently, thereby emphasizing their (lack of) completeness and the (lack of) trust bestowed on them. Together, these were amongst the prime areas considered to which a data-mining solution could be applied.

Exploring the data is an important early step within any data-mining process, during which the data's characteristics, their quality, and their abnormalities are inspected. Generally, a generous amount of data is provided from which conclusions can be drawn that influence choices made during the development of the eventual data-mining solution. Unfortunately, the minimal amount of data currently available through ARIADNE prevents such a sequence of events to take place. Therefore, Linked Data from several different archaeological repositories around the globe was inspected instead. These data were chosen for their almost-disjoint characteristics, thus hopefully providing good representations of the different facets that ARIADNE might bring forth. Assuming they do, several observations could be made: save for the generally expected differences in used ontologies and structure, the examined linked archaeological data were found to strongly depend on descriptive values, as well as consisting largely of relatively flat data structures. These aspects of the data should be considered during the development of the forthcoming data-mining solution.

Understanding the domain and its data are two early but crucial steps in any data-mining process. Together with our updated knowledge on Linked Data and data mining, these all come together to form the field of Semantic Web mining. This field represents a young area of research of which many aspects are still uncertain or left unexplored, from both a technical and practical perspective. In fact, many of methods are still under heavy development, with few of them having progressed outside the confines of academic research. Therefore, instead of considering all possible approaches, we have solely focussed on the more-prominent movements as seen in the literature.

Recommendations

Generally, the developer of a typical data mining solution will explore a large amount of data with the goal of revealing potentially relevant patterns. After careful inspection, the truly relevant patterns will subsequently be generalized to the entirety of the data. Unfortunately, the small amount of data currently available through ARIADNE would make it rather unlikely to successfully generalize about any discovered pattern for the large amount of data that, one day, will be accessible. Instead, a more generic approach is suggested, such that its workings are ensured regardless of the exact characteristics of the future data.

Based on the study of both domain and data, as well as on practical constraints with respect to time and resources, two data-mining solutions were chosen which were deemed the most feasible and suitable for implementation within the ARIADNE infrastructure. These constitute 1) the ability for users to generate potentially-relevant hypotheses, and 2) analysing the quality of data as well as helping to improve it. We will briefly touch on these two solutions next.

Hypothesis Generation (7.1)

The official project proposal of ARIADNE mentions the ability to detect patterns in archaeological data or related data, and applications within the ARIADNE infrastructure. Data mining methods are capable of detecting such patterns. Interesting and potentially relevant subsets of these patterns can then be presented to users as starting points for forming new research hypotheses. The researcher might already have a hypothesis, or the patterns may reveal something the researcher is interested in exploring further. The interestingness of patterns will be generated algorithmically on the basis of predefined criteria and user feedback. To facilitate this, a user interface should provide access to a data-mining backend. Initially, this might best be integrated within a text-based query interface such as SPARQL or similar. At a later stage, a wrapper into the graphical user interface should be made. This will allow multiple hypotheses to be presented in non-intrusive ways, thereby allowing users to quickly scan for potentially valuable directions.

Data Quality Analysis (7.5)

Two aspects that reflect poorly on the quality of data are the occurrence of gaps and errors in the knowledge contained therein. In case of the former, filling these voids involves predicting the most-likely resource, link, or literal. In the latter case, these errors typically contain anomalies within the data, which cannot be explained by any of the discovered patterns alone. Depending on the likelihood of them being erroneous, the detected errors could be suggested for removal or tagged as dubious. Alternatively, they could be replaced by a prediction on the correct value. This could reuse the prediction method mentioned earlier, and data mining backend.

Roadmap

The sequel to this report, i.e. Deliverable 16.3, will present the final results of the applicability and feasibility of data mining within the ARIADNE infrastructure. To this end, the aforementioned recommendations will be explored and experimented with further. This process will consist of several phases.

Following this report, a more extensive study into the recommended topics will first be performed. This will assist in narrowing down the list of possible options to only those we believe to possess the most potential. The remaining options will subsequently be implemented into our experimental environment on-site, within which they will thoroughly be tested on various linked archaeological data. The result of these tests will determine whether the selected options are suitable for integration within ARIADNE. Based on the experience gained during the current study, we expect that most of the possible options will need to be adapted to suit both the data and the users' needs. If, for some reason, none of these options are found to be suitable, a custom solution will be developed instead.

Once the selected options have successfully been implemented within the experimental environment, internal evaluation rounds will be organized during which domain enthusiasts and experts of varying levels of expertise will be asked to experiment with the implementation. Here, we expect the groups to

likely be comprised of students, junior and senior archaeological researchers, and local data and repository managers. This will additionally provide the input needed for the development of (elements of) a graphical user interface. Similar to what has been done before; an iterative scheme will be followed.

The final phase will consist of implementing the data mining solutions into the ARIADNE infrastructure. This is followed by extensively experimenting on various data accessible through ARIADNE. This implementation will be improved upon further during a series of iterative and open evaluation sessions for the remainder of the ARIADNE project.

1 Introduction and Objectives

ARIADNE, the Advanced Research Infrastructure for Archaeological Dataset Networking in Europe, will facilitate a central web portal that provides access to archaeological data from various sources in a standardized and open format. This format will likely adhere to the Linked Data paradigm either fully or partially, with the former being the option that we believe is needed to propel ARIADNE towards a higher level of interoperability. By this assumption, users will be able to use the portal to browse and search the data, thereby making use of all the advantages that Linked Data has to offer. Among these advantages are advanced search abilities, the inherent capability of drawing inferences, and the enrichment of data by linkage to and from external sources. We expect these features to have a positive impact on the archaeological research community. However, this does not necessarily add to the knowledge contained within the aggregated data sets. Ideally, we would like to expand on this knowledge as well. One field of expertise that specializes in exactly that is **data mining** (Hastie, et al. 2009).

Both Linked Data and data mining are relatively innovative approaches, which have only recently started to prove their usefulness outside the confines of academic research and, more specifically, Computer Science. As a result, the average archaeologist has little to no experience with either of these approaches and is often unaware of what they actually entail (Selhofer and Geser 2014, Hollander and Hoogerwerf 2014). Fortunately, their interest within the field has experienced a rise during the last decade, with both being covered periodically at conferences and workshops of Computer Applications and Quantitative Methods in Archaeology (CAA) (Doerr, Schaller and Theodoridou 2004, Isaksen, Earl, et al. 2009, Earl, Sly and Wheatley 2014). Even so, it would be prudent to begin this deliverable with a short introduction into both Linked Data and data mining, to inform the less-familiar reader about this exiting field. Moreover, a basic understanding of both fields is required to grasp the subsequent section about data mining *on* Linked Data.

1.1 Structure of Report

Following this section, a short introduction into Linked Data and data mining will be made, during which several relevant core concepts will be discussed. This is followed by a technical examination of how data mining may be applied to Linked Data, thereby considering relevant literature, applications, and experiences within this field. Once all the theoretical topics have been covered, we will concentrate on how this would fit within the context of ARIADNE. Hereto, we will thoroughly examine the users' wishes and requirements, as well as analyse the currently available data. All these topics merge in the next-to-last section, in which we will look at the applicability of data mining on Linked Archaeological Data in ARIADNE. Finally, a conclusion will be formed and recommendations will be presented.

2 Introduction to Linked Data

Over the last few decades, the World Wide Web has radically changed our way of sharing knowledge (Shadbolt, Hall and Berners-Lee 2006). Until recently however, this knowledge was mostly contained within documents, such as a web page, an image, or a postscript file (Bizer, Heath and Berners-Lee 2009). Typically, these documents are unstructured or semi-structured at best, with the former being nothing more than a dump of symbols and the latter being structured data with a very loose or no formal schema (Buneman 1997). An example of this latter category is XML (Bray, et al. 2008), of which their schema is largely contained in the corresponding data themselves.

Sharing knowledge through unstructured and semi-structured documents has several limitations. These concern the reusing and integration of data, as well as the discovery of relevant knowledge contained in that data. These limitations may be mitigated by using a structured standard with predefined semantics (Heath and Bizer 2011, van Harmelen, et al. 2012). Such structure lies at the heart of Linked Data[1] (LD).

Envisioned and recommended by the World Wide Web Consortium[2] (W3C), LD is a paradigm that strives to bring forth a set of best practices to store, share, and interpret data on the web. A key aspect of this is the links between the different data throughout the web. By connecting these with each other, a web of data emerges through which researchers can search, and enables those researchers to analyze vast amounts of data. Moreover, it allows the knowledge contained in that data to be interpretable by both humans and machines, thereby opening up the path to answering more complex questions. For instance, combining data on several archaeological excavations with data on biology, geography, and carbon dating could become a more trivial task through using LD practices.

Linked Data is one of the key components to be explored within ARIADNE. Therefore, in order to investigate how data mining may apply within ARIADNE, we first need to understand the data with which it is supposed to work. Hence, we will now foray into the world of LD where we will discuss the elements deemed relevant for the discussion on data mining (DM).

2.1 The RDF Data Model

The data model on which LD is built and to which DM will be applied is called the **Resource Description Framework** (RDF) (Bizer, Heath and Berners-Lee 2009, van Harmelen, et al. 2012, Shadbolt, Hall and Berners-Lee 2006). This framework provides a powerful and simple method to specify information with in the form of binary statements. These statements are specified using the RDF Data Model. This model consists of four core concepts; resources, properties, statements, and graphs.

[1] Linked Data, see www.w3.org/standards/semanticweb/data
[2] World Wide Web Consortium, see www.w3.org

2.1.1 Resources

Resources can be thought of as the entity about or with which a statement is made. These can be almost anything imaginable, be it a person, a book, a number, a theory, or anything in between. In order to minimize any ambiguity between resources, e.g. when its meaning differs per domain, each resource has an **Universal Resource Identifier** (URI) (van Harmelen, et al. 2012, Heath and Bizer 2011, Stumme, Hotho and Berendt 2006). This URI is used to refer to that specific resource only. Therefore, each URI is unique; any statement that uses that URI means exactly the piece of knowledge contained in the corresponding resource. In other words, two statements using the same URI are expected to mean exactly the same thing. For instance, such a URI will make it possible to easily distinguish between an *archaeological context* and a *literary context,* when only the term *context* is used.

A URI takes the form of a web address (van Harmelen, et al. 2012, Heath and Bizer 2011, Bizer, Heath and Berners-Lee 2009), e.g. *https://www.example.org/Italy* to refer to a resource about the country of Italy. Often however, URIs make use of the *fragment identifier* denoted by the hash symbol (#), e.g. *https://www.example.org/country#Italy* and *https://www.example.org/country#Germany*. Note that these resources do not necessarily need to exist, as is the case with many physical entities. In fact, it is considered good practice to create URIs for these entities. A prime example of this are (non-digitalized) books, which are referred to by ISBN number. Another, less straightforward, example are people themselves.

While URIs are well suited for linking to clearly defined and bounded concepts, they are less able to cope with those that are less so. Within this latter category reside numerical and textual properties of which the precise description is difficult to capture in a single relation. These entities are referred to as *literals* (van Harmelen, et al. 2012, Heath and Bizer 2011, Bizer, Heath and Berners-Lee 2009). As an example, consider specifying the depth at which an artefact was discovered. Depending on the precision, the number of required resources needed to enable all possible values would approach infinity. Instead, a single numerical value would typically be used.

2.1.2 Properties

At the heart of the RDF data model lie the relations between its resources (van Harmelen, et al. 2012, Heath and Bizer 2011, Bizer, Heath and Berners-Lee 2009). These relations are described by properties. Note that the data model supports only binary relations and that all relations are unidirectional. That is, *resource A* is related to *resource B* by some property. For instance, an artefact is related to an excavation by a property that denotes that the former is found at the latter.

2.1.3 Statement

An RDF statement constitutes the **triple** (*subject, predicate, object*), whereby the *subject* is related to the *object* as specified by the *predicate* (

Figure 2-1) (Heath and Bizer 2011, Bizer, Heath and Berners-Lee 2009, van Harmelen, et al. 2012). Here, the *subject* is a resource, the *predicate* a property, and the *object* is either a resource or a literal. As an example, assume the triple (*Dragendorff 33, has dating, 200*) which uses a *predicate* to state that the *subject* has been dated as described by the *object*. Here, positive values are assumed to be *AD*, while negative values are assumed *BCE*.

Figure 2-1: RDF data model. The subject is related to zero or more objects by zero or more predicates.

2.1.4 Graphs

While RDF statements are generally stored as triples, they fit more naturally as a labeled and directed graph. Each node on such an RDF graph is a resource, with the connecting arcs denoting the properties between the nodes (van Harmelen, et al. 2012, Heath and Bizer 2011). Hence, a single RDF statement constitutes a graph with two interlinked nodes.

By allowing resources to be linked, ever-larger growing graphs can be constructed (Heath and Bizer 2011). As a simple example (Figure 2-2), consider the triple (*Item #42*, *is an*, *Artefact*) as might be stored in an archaeological database. Furthermore, assume the *subject* to additionally be part of the triples (*Item #42, of type*, *Dr 33*) and (*Item #42, found at depth*, *-1.32*), with the latter containing a literal as *object*. Finally, assume the triple (*Dr 33*, *has dating*, *200*), thereby using *Dr 33* as *subject*, whereas it was previously used as *object*. The resulting graph will contain three resources, four properties, and two literals.

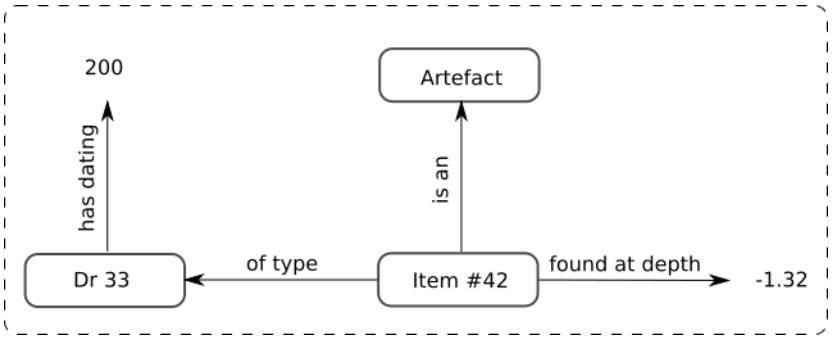

Figure 2-2: Schematic depiction of a simple RDF graph.

2.2 Ontologies

As discussed previously , RDF is a simple data model with which statements can be specified. In fact, RDF alone can be used to make *any* statement, even ludicrous ones such as that the Pyramids of Giza are in Rome. The reason behind this is that it does not make any assumptions about the domain it describes, nor the semantics used to describe that domain. That is, RDF itself is "unaware" of the information it states. Instead, this "awareness" is provided by *ontologies* (Shadbolt, Hall and Berners-Lee 2006, van Harmelen, et al. 2012, Heath and Bizer 2011).

Within the field of Computer Science, ontologies describe a domain by types, properties, and relation types. They are the culmination of a progression from simple vocabularies with a fixed list of terms to full-fledged languages with a powerful expressiveness (Garshol 2004, van Harmelen, et al. 2012). To this end, ontologies generally provide a hierarchy of classes and properties, as well as allowing some form of reasoning over them. These classes may encompass *subjects*, *predicates*, and *objects*, including both *resources* and *literals*. For instance, an archaeological ontology might define the exact meaning of the concept *excavation*, and that this concept should always contain an archaeological context to be valid.

Numerous ontologies, both simple and powerful, modelling various domains have already been made available in an RDF-compliant format. Within ARIADNE, the more relevant of these is, unsurprisingly, the archaeological domain. As this domain has a strong geospatial component (Wagtendonk, et al. 2009, De Kleijn, et al. 2014, Conolly and Lake 2006), this latter domain can be regarded as quite relevant as well. Therefore, several ontologies describing these two domains will be briefly discussed next. Please note that, for simplicity, the technical aspects behind ontologies have been omitted.

2.2.1 Geospatial Ontologies

There exist several ontologies holding geospatial knowledge, of which the most elementary is the Basic Geo Vocabulary (BGV) (Brickley 2006). BGV allows for the specification of points within the World Geodetic System (WGS) standard. To this end, it accepts latitude, longitude, and altitude declarations.

A geospatial ontology with a more-powerful level of expressiveness than with BGV is GeoSPARQL; a geographical query language developed by the Open Geospatial Consortium[3] (OGC) (OGC GeoSPARQL - A Geographic Query Language for RDF Data 2012). While it emphasizes the retrieval of knowledge, i.e. querying, more than describing, GeoSPARQL still offers a flexible ontology to describe topologies. To accomplish this, it incorporates specifications from other geospatial standards designed by the OGC, among which are the Geographic Markup Language[4] and Simple Feature Access[5]. Therefore, GeoSPARQL accepts declarations of points, (multi) lines, and (multi) polygons, as well as describing their properties

[3] Open Geospatial Consortium, see www.opengeospatial.org
[4] Geospatial Markup Language, see www.opengeospatial.org/standards/gml
[5] Simple Feature Access, see www.opengeospatial.org/standards/sfa

by Region Connection Calculus. However, some of the more exotic features require that the data store supports the GeoSPARQL protocol.

2.2.2 Archaeological Ontologies

An ontology within the archaeological domain will be the ARIADNE Catalogue Data Model (ACDM) (Aloia, et al. 2014). The ACDM attempts to provide a data model to describe archaeological resources, such as collections, data sets and services, as well as metadata and vocabularies. Due to this area of focus, it is being built upon the Data Catalogue Vocabulary (DCAT) (Maali, Erickson and Archer 2014); a vocabulary commended by the W3C for its ability to represent government data catalogues. Instead of catalogues however, ACDM emphasizes collections and data sets, with the former being a set of heterogeneous items without a formal structure and the latter being a set of structured records. These structured records are assumed to originate from either a database or from a Geographic Information System (GIS).

Two other archaeological ontologies are the CRMarchaeo and CRM-EH extension of the CIDOC Conceptual Reference Model; an ontology for describing knowledge from the domain of cultural heritage (The CIDOC Conceptual Reference Model n.d., Doerr and Schaller 2008). The CIDOC CRM was developed by in collaboration with the International Council of Museums, with the aim of allowing diverse perspectives that incorporate different institutional histories, disciplines, and objectives. To this end, it provides a solid core with the ability to add functionality by use of extensions.

The CRMarchaeo constitutes a generic archaeological extension, developed within the ARIADNE framework, which aims at encoding metadata on the excavation process (Cripps, et al. 2014). By offering this metadata, CRMarchaeo endeavors to optimize the interpretability of a documented excavation, thereby providing the rational for conducting that excavation, as well as knowledge on previous excavations and studies on the same site.

The second archaeological CRM extension, the CIDOC CRM-EH (May n.d.), was developed to include the archaeological concepts and processes in use by the English Heritage[6]; a national heritage body in the UK charged with safeguarding cultural heritage. To this end, it offers numerous classes and (inverse) properties divided amongst several modules.

2.3 The Semantic Web

The LOD cloud constitutes a large number of interconnected RDF repositories. These repositories, commonly referred to as a **triple stores**, allow queries to be processed on their data (Shadbolt, Hall and Berners-Lee 2006, van Harmelen, et al. 2012).

[6] English Heritage, see www.english-heritage.org.uk

A single, isolated triple store already provides a data structure that can allow powerful methods of searching through and reasoning with the data to be used. The real advantage of LD however, only surfaces when multiple triple stores are available on the web and are linked to each other. Recall that this distinction is equivalent to the difference between the four and five star rating of the LOD project as was discussed earlier. Irrespective, these interlinked triple stores together form a **web of data**; the *Semantic Web* (Bizer, Heath and Berners-Lee 2009, Heath and Bizer 2011, van Harmelen, et al. 2012).

2.3.1 Querying the Semantic Web

Nearly all search engines for the current-day World-Wide Web are keyword-based (Freitas, et al. 2012); given a provided set of words the most relevant result is sought. Generally, this is accomplished by some variation on **Vector Space Models** (Appendix B), which constitute a method with which documents can be identified. Unfortunately, such a solution would be unsuited for searching through the LOD that makes up the Semantic Web (SW) or web of data, as it lacks the ability to represent structured data as well as their semantics (Figure 2-3). Instead, the SW is generally searched through by query-based languages that were specifically developed for this purpose.

In the case of a triple store, the commonly used query language is that of **SPARQL** (Prud'hommeaux and Seaborne 2008, van Harmelen, et al. 2012, Heath and Bizer 2011, Shadbolt, Hall and Berners-Lee 2006, Bizer, Heath and Berners-Lee 2009). SPARQL is a W3C recommended protocol and a query language developed to access, retrieve, and modify RDF data. Similar to other query languages, such as SQL, it offers a wide range of capabilities ranging from simple pattern matching to complex queries with restrictions in range, time, and domain. Moreover, SPARQL's query-processing engine enables the ability to reason deductively (Appendix A) over the data to which it provides access. This ability can be either relatively powerful or rather limited, depending on the expressive strength of the ontologies used to specify these data with.

Querying a triple store through a query-language like SPARQL is accomplished by connecting to a so-called *endpoint*. Endpoints are provided by the triple store and form the bridge between a user and the data contained in that triple store (van Harmelen, et al. 2012, Heath and Bizer 2011). Once a query has been submitted to such a publically accessible endpoint, it will be processed and the results returned. For instance, a query might request all pottery fragments with burn marks and larger than 20 cm^2 that were found between 1950 and 1960 at a specific site in Italy.

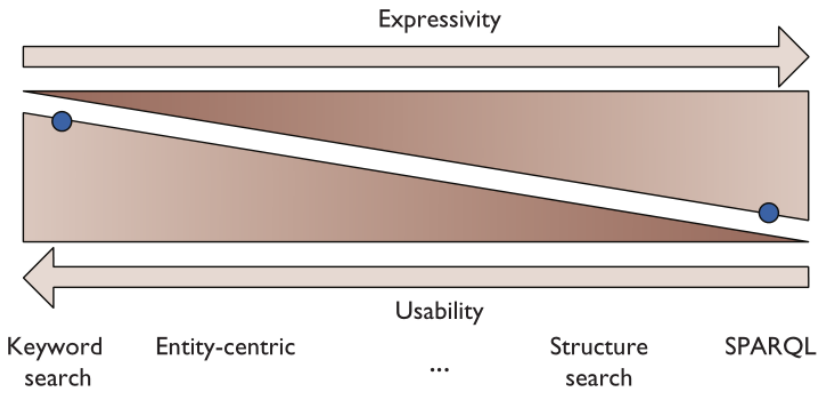

Figure 2-3: Expressivity-Usability trade-off for querying with either a word or ontology-based search engine (Freitas, et al. 2012).

2.4 Linked Archaeological Data

While fairly new, the concept of LD is not unheard of within the archaeological domain. In fact, some think of it as being the next logical step in sharing archaeological knowledge (Signore 2009, Richards 2006). Others believe however its semantic complexity lacks the ability to describe the uncertainty in archaeological data (Isaksen, Martinez, et al. 2010, Martinez and Isaksen 2010). Regardless, several endeavors regarding Linked Archaeological Data (LAD) repositories have already been undertaken.

One of the larger LAD repositories within Europe is disseminated by the Archaeology Data Service[7] (ADS), which provides a SPARQL endpoint to a triple store hosted by the University of York[8] (Charno, et al. 2012). This triple store was developed as part of the STELLAR[9] project (Tudhope, et al. 2011); a collaboration, funded by the UK Arts & Humanities Research Council[10], in partnership with the University of Glamorgan (now South Wales)[11], and English Heritage, with the aim of improving the integration of LD into the digital archaeological domain. The data currently in the ADS triple store were converted from databases and spreadsheets to RDF, using the CRM-EH ontology. The resulting triples are stored in an AllegroGraph[12] triple store, which points to a SPARQL endpoint and allows the results to be provided in one of several RDF serializations.

On the other side of the Atlantic ocean, the Digital Index of North American Archaeology (DINAA) aims to integrate government curated public data from both offline and online digital archaeological

[7] Archaeological Data Service, see data.archaeologicaldataservice.ac.uk
[8] University of York, see www.york.ac.uk
[9] STELLAR Project, see www.archaeologicaldataservice.ac.uk/research/stellar
[10] Arts & Humanities Research Counsel, see www.ahrc.ac.uk
[11] University of Glamorgan, see www.southwales.ac.uk
[12] AllegroGraph, see www.franz.com/agraph/allegrograph

repositories (Wells, et al. 2014). Supported by the National Science Foundation[13], its primary focus concerns aiding the researcher in data discovery, as well as filling the gap in archaeological information infrastructures. Moreover, DINAA emphasizes the reuse of both technologies and data. These data, which are stored in a MySQL[14] database, are expressed with the help of the DINAA[15] vocabulary; an ontology built upon OWL and strongly-influenced by CIDOC CRM. Furthermore, as these data are open, DINAA has been welcomed into the LOD cloud.

Where both ADS and DINAA provide national data for the most part, ARIADNE will attempt to extend this to the whole of Europe. Not all LAD projects aim at such scale however. For instance, (Gruber, et al. 2012) explore the feasibility and usefulness of introducing LD into the field of numismatics, thereby providing enhanced searching abilities to a database of Roman coins. Another example is from (Isaksen, Martinez, et al. 2009), who tried to improve our understanding of ancient trade networks by analyzing LAD concerning the distribution of amphorae and marble. As a final example, consider the research done in de Boer, et al. (2014) where they paired LAD concerning Dutch ship wrecks with that of Dutch sailors, resulting in new insights on the socio-economic realities of the 18th Century.

[13] National Science Foundation, see www.nsf.gov
[14] MySQL, see www.mysql.com
[15] DINNA Ontology, see opencontext.org/vocabularies/dinaa

3 Introduction to Data Mining

Data mining (DM) is a fairly new and multi-disciplinary field which intersects with Artificial Intelligence, Data Science, and Statistics, as well as partially overlapping with Machine Learning (ML) from which it draws its technical basis (Kantardzic 2011, Hastie, et al. 2009, Friedman 1998, Witten, Frank and Hall 2011). Therefore, the people who specialize in this field stem from various backgrounds and hold different views, making it difficult to provide a definition agreed upon by all those involved. Consider the following three definitions, as found in the literature:

> *Data Mining it the nontrivial process of identifying valid, novel, potentially useful, and ultimately understandable patterns in data* (U. M. Fayyad 1996).
>
> *Data Mining is the extraction of implicit, previously unknown, and potentially useful information from data* (Witten, Frank and Hall 2011).
>
> *Data Mining is a decision support process where we look in large data bases for unknown and unexpected patterns of information* (Parsaye 1996).

A common notion throughout the different definitions is that of identifying, extracting, and using information from data that was previously unknown. In order words; Data Mining concerns **learning from data** (Hastie, et al. 2009).

3.1 Learning from Archaeological Data

Long before the field of DM came to be, statistics was the only area that specialized in learning from data. Despite its potential usefulness however, many archaeologists used to refrain from familiarizing themselves with these methods (Baxter 2003). Hence, their use in the archaeological domain progressed rather slowly. Instead, most grew out of necessity, i.e. to solve a problem, and were originally developed in other fields such as geography and ecology. Only with the emergence of *New Archaeology*, did the interest in statistical methods grow. Still, early uses were largely without the then-called 'complex statistics', amongst those listed were principal component (PCA), factor, and cluster analysis (Whallon 1987, Kintigh 1987). This changed with the increased availability of statistical applications.

Over the last two decades, a rise in computational power gave birth to various statistical applications (Baxter 2003). Some of these, such as SPSS[16] and, in a lesser degree, R[17], were aimed at individuals who were not really familiar with the theory behind statistical methods. These tools allowed the archaeologists to apply and get acquainted with basic approaches such as regression analysis and Bayesian statistics, as well as providing a simple frontend to the aforementioned 'complex statistics'.

Recent years saw the integration of statistical methods into various non-dedicated applications. For example, consider a Geographical Information System (GIS); a tool commonly used by archaeologists to perform some form of spatial analysis (Selhofer and Geser 2014, Baxter 2003). Hereto, most modern GIS frameworks provide a simplified frontend to several adapted statistical methods, thereby including location and predictive modelling, as well as a subset of the 'complex statistics' mentioned earlier.

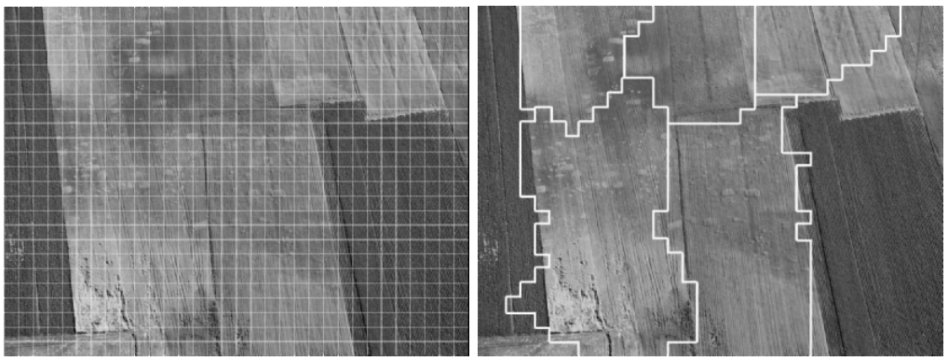

Figure 3-1 : Rough segmentation of an archaeological aerial photograph as determined by a DM algorithm (Kobylinski and Walczak 2006).

The acceptance of DM by archaeologists appears to follow a line similar to that of statistics, with the term "data mining" having been mentioned only sporadically in archaeologically-related literature. Nevertheless, certain topics related to DM appear to be quite well represented, especially those involving some form of classification. Amongst these, the often-encountered artefacts are coins, glass, and ceramics, which are classified based on the similarity between their visual characteristics (van der Maaten, et al. 2006, Huber, et al. 2005, Nolle, et al. 2003, Karasik, et al. 2004). An example of a more specialized study involving classification is that of Bi, et al. (2008), who created a method to spatially classify and partition archaeological settlements based on the discovery of nearby hearths, pits, urn tombs, and pit tombs. Another specialized example is the research conducted by Linderholm & Geladi (2012), who developed an approach to classify archaeological soil and sediment samples based on infrared readings of those samples. As yet another example, consider Di Ludovico and Pieri (2011), who explored various means to classify entries within large corpora of decorations on Mesopotamian cylinder seals. As a final example, consider the research done by Kobylinski & Walczak (2006), who

[16] SPSS, see www.ibm.com/software/nl/analytics/spss
[17] The R-project, see www.r-project.org

developed a method to automatically determine potentially interesting features in archaeological aerial photos (Figure 3-1).

3.2 Knowledge Discovery and Data Mining

Until now, we have used the term "data mining" to refer to the whole process of discovering useful patterns from any form of data. For simplicity sake, we will continue to do so. Strictly speaking however, the term merely denotes the act of running an algorithm on a data set. This is only one stage in a larger knowledge discovery process, generally known as Knowledge Discovery and Data Mining (KDD). During the course of ARIADNE, such a process will be undertaken by the knowledge engineers involved with WP 16. In fact, the research being conducted for this report already constitutes a partial KDD process.

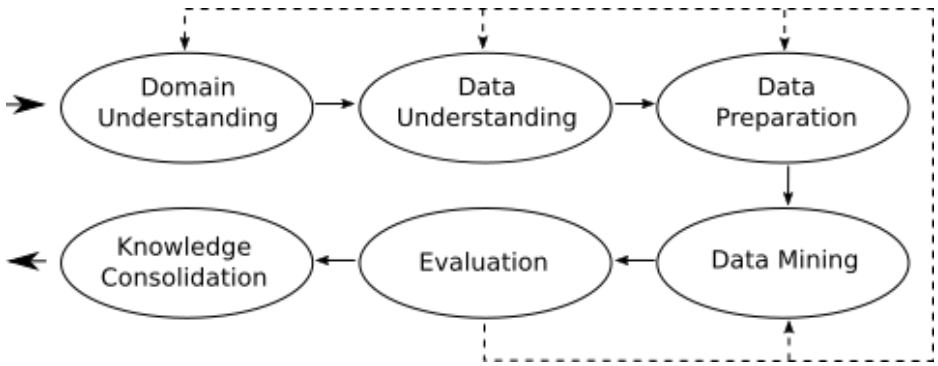

Figure 3-2: Schematic depiction of a generic KDD process.

A generic KDD process (Figure 3-2) typically consists of six different stages (Kurgan and Musilek 2006): Domain Understanding, Data Understanding, Data Preparation, Data Mining, Evaluation, and Knowledge Consolidation. For each of these, a general description will be given.

Domain Understanding concerns familiarizing oneself with the domain at hand (Fayyad, Piatetsky-shapiro and Smyth 1996, Kantardzic 2011, Kurgan and Musilek 2006, Maimon and Rokach 2005). This entails a sufficient comprehension of the current state of affairs, on the problems therein, and on the goals that have to be reached. Furthermore, key figures and their terminology should be identified. Within ARIADNE, this boils down to an understanding of the archaeological domain, as well as of the archaeologists themselves.

Data Understanding concerns analysing the data, thereby inspecting its quality (Witten, Frank and Hall 2011, Fayyad, Piatetsky-shapiro and Smyth 1996, Kantardzic 2011, Kurgan and Musilek 2006). This involves the identification of anomalies, such as noise, outliers, and missing values, as well as the selection of interesting subsets or features. In the case of ARIADNE, this involves an understanding of LAD and its anomalies.

Data Preparation concerns the transformation of the data as to make it suitable for DM (Witten, Frank and Hall 2011, Fayyad, Piatetsky-shapiro and Smyth 1996, Kantardzic 2011, Kurgan and Musilek 2006, Maimon and Rokach 2005). This entails resolving problems in data quality, which were discovered previously, as well as scaling and normalizing values if needed. Furthermore, a final selection of interesting features is made.

Data Mining concerns applying a suitable inductive-reasoning method (Appendix A) to the prepared data set, resulting in the automated discovery of potentially relevant patterns (Witten, Frank and Hall 2011, Fayyad, Piatetsky-shapiro and Smyth 1996, Kantardzic 2011, Kurgan and Musilek 2006, Maimon and Rokach 2005). These patterns are described in a mathematical model that approximates the data.

Evaluation is the phase during which the previously generated knowledge is interpreted, as well as being inspected for its usefulness (Witten, Frank and Hall 2011, Fayyad, Piatetsky-shapiro and Smyth 1996, Kantardzic 2011, Kurgan and Musilek 2006, Maimon and Rokach 2005). This typically involves a visualization of the corresponding patterns. In the case of ARIADNE, this will involve an iterative review process attended by both knowledge engineers and archaeological researchers.

Knowledge Consolidation concerns the presentation of the new knowledge in a user-oriented fashion, followed by the possible incorporation of that knowledge into a final system (Witten, Frank and Hall 2011, Fayyad, Piatetsky-shapiro and Smyth 1996, Kantardzic 2011, Kurgan and Musilek 2006, Maimon and Rokach 2005). Within ARIADNE, this would correspond to either showing the DM results to the user or adding these results to the corresponding triple store.

The generic sequence of stages as outlined above is only one of many proposed KDD models (Kurgan and Musilek 2006). Due to its domain-independent properties, we believe it to be a suitable model to follow during our research within this WP. In fact, the steps of Domain and Data Understanding will be covered largely during the course of this report.

3.3 Data Mining Tasks

The domain-understanding stage of a KDD process typically provides the knowledge engineer with insight into the desired goals of that process. These goals form the main criteria when deciding on which DM task to implement. While many variants exist, these tasks generally fall into one of the following higher-level tasks (Fayyad, Piatetsky-shapiro and Smyth 1996, Kantardzic 2011, Lavrac and Dzeroski 2001) : Classification, Regression, Clustering, Summarization, Change and Deviation Detection, and Dependency Modelling. For each of these, a general description will be given.

Classification focusses on learning a predictive model that is capable of correctly assigning new instances of unknown classes to one of several predefined classes (Fayyad, Piatetsky-shapiro and Smyth 1996, Witten, Frank and Hall 2011, Kantardzic 2011, Lavrac and Dzeroski 2001, Berendt, et al. 2004, Hagood 2012). Typically, these classes represent related categories within a certain domain, for example, *red* and *green* denote classes within a finite set of colours.

Regression analysis involves learning a model, which can predict unknown attribute values based on known values of the other attributes belonging to the corresponding instances (Fayyad, Piatetsky-shapiro and Smyth 1996, Witten, Frank and Hall 2011, Kantardzic 2011). Here, both known and unknown values should have a numerical internal representation. In the case of binary or categorical values, e.g. labels, specific numerical ranges are used. For instance, consider predicting either *true* or *false* by letting a positive and negative value denote the former and latter, respectively.

Cluster analysis tries to describe a finite set of groups or clusters composed of instances with similar attribute values (Fayyad, Piatetsky-shapiro and Smyth 1996, Witten, Frank and Hall 2011, Kantardzic 2011, Lavrac and Dzeroski 2001, Berendt, et al. 2004, Hagood 2012). These clusters are determined without prior knowledge on the underlying structure of the data, and may be regarded as nameless classes. Hence, cluster analysis can be seen as a variant of classification.

Summarization concerns the methods that are capable of compressing data into more compact forms without losing too much of the original knowledge (Fayyad, Piatetsky-shapiro and Smyth 1996, Kantardzic 2011, Lavrac and Dzeroski 2001). This problem can be tackled through two distinct paths; either through *extraction* or through *abstraction* (Mani and Maybury 1999). Here, the former entails the automatic extraction of existing fragments of the data that are deemed relevant, while the latter method generates new data that describes these relevant aspects in a concise way. The alert reader might recognize the similarities between the method of *abstraction* and the technique behind VSM.

Change and Deviation Detection aims at discovering significant changes or deviations, e.g. outliers, from previously measured or normative values, respectively (Fayyad, Piatetsky-shapiro and Smyth 1996, Kantardzic 2011, Lavrac and Dzeroski 2001). In both cases, the key is to determine whether the probability of such an anomaly occurring is too low to warrant it to actually happen. For instance, given an average human height of 1.70 meters, the arrival of someone with a height of 2.50 meter would almost certainly stand out.

Dependency Modelling consists of learning one or more models that are capable of describing significant dependencies between the different variables found in the data (Fayyad, Piatetsky-shapiro and Smyth 1996, Kantardzic 2011). Typically, these models focus on a particular subset of the data, thereby describing different dependencies, and thus do not cover the entire data set.

None of these six high-level tasks make assumptions on the domain at hand, nor do any of the algorithms used to perform these tasks. They do however, make assumptions on the data to which they

are being applied. Within ARIADNE, these data constitute LD as found on the SW. Hence, algorithms should be used which were developed for mining the Semantic Web.

3.4 Towards Mining the Semantic Web

The field of DM originated at a time during which a two-dimensional table containing features and instances (Figure 3-3 Left) was the most-commonly used format in which to digitally store information (Kantardzic 2011, Witten, Frank and Hall 2011, Knobbe 2006). Hence, most research and development within the field focussed on tabular data. As a result, this area of DM gained much experience over the past few decennia. It is this form of DM that is generally referred to by the term “data mining”.

The simplicity behind the tabular format limits the inference capabilities over data stored as such. In terms of logic, these limitations prevent one from inferring anything other than simple statements. Within the field, these statements are known as propositions. Hence, tabular data and DM thereon are commonly referred to as propositional data and **propositional data mining**, respectively (Berendt, Hotho and Stumme 2002, Chen, Han and Yu 1996). Statistics, due to its large and proven experience with propositional data, typically forms the basis for many of the methods used in this part of the DM field. A direct consequence of this inheritance is the assumptions made about the data to which these methods are applied. Most important is the assumption that the values in the data set are independent of each other; changing or leaving out one value should have no repercussions for any of the other values. The validity of any discovered pattern would be doubtful if this assumption was found to be false. Fortunately, it tends to hold up for tabular data.

The limitations and assumptions inherent to the tabular format make it unsuitable for storing more complex forms of data. Moreover, the sheer volume of some data sets would make it rather impractical to work with if they were stored in a single table. A fairly straightforward solution would be to split the data amongst multiple related tables. This is the general idea behind the format used to store data in relational databases (Figure 3-3 Centre).

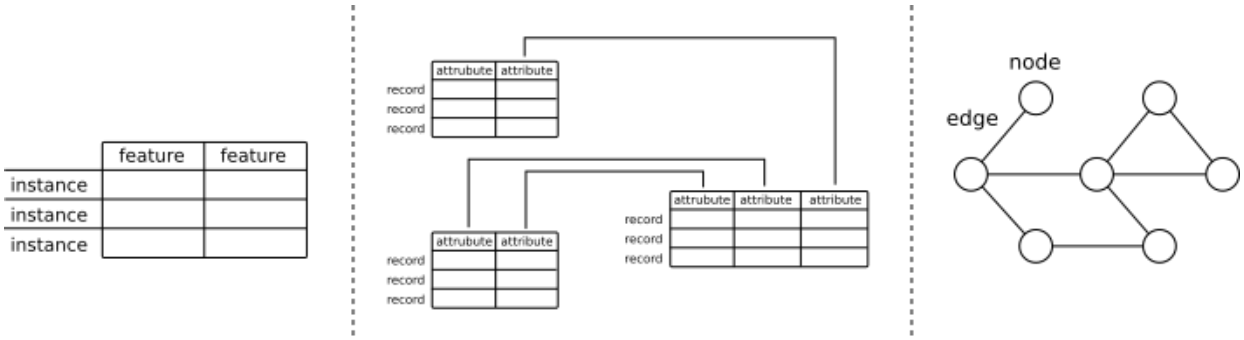

Figure 3-3: Three types of data to which DM may be applied. Left) Propositional data. Centre) Relational Data. Right) Graph Data.

Relational databases implement the Relational Data Model (Codd 1970). This model allows for data to be split up amongst one or more table-like structures. Each of these structures represents a distinct relation between its attributes, e.g. an archaeological context. Moreover, each attribute may occur in more than one structure, thereby being linked through a unique key. Due to these characteristics the assumption of independent values no longer holds. Hence, propositional DM is unsuitable for this form of data. This realization led to a whole new branch of DM known as **(Multi-) Relational Data Mining** (MRDM); an area of expertise that falls under Knowledge Discovery in Databases (Maimon and Rokach 2005, Lavrac and Dzeroski 2001, Knobbe 2006).

The Relational Data Model can be seen as the intermediate step between propositional data and graph data (Figure 3-3 Right). While the difference between the Relation Data Model and graph data is still considerable, both acknowledge the possible dependencies between their respective elements. As RDF is a graph-based data format, experiences from the MRDM field may thus provide valuable insights into DM on RDF graphs. These insights, together with relevant techniques and applications, contribute to a DM branch known as **Semantic-Web Mining** (SWM) (Rettinger, et al. 2012, Stumme, Hotho and Berendt 2005). As evident by its name, this branch specializes in DM on the SW.

4 Semantic-Web Mining

Semantic-Web Mining (SWM) is an umbrella term which denotes the area of DM that focusses on mining LD as found on the SW. It is a relatively new area of research of which many aspects are still uncertain or left unexplored, from both a technical and practical perspective. If fact, many of its tasks and learning methods are still under heavy development, with few of them having progressed outside the confines of academic research. Instead of presenting an exhaustive list of all possible approaches, we will therefore solely focus on the more-prominent movements as seen in the literature. Please note that, for simplicity reasons, this section will refrain from discussing any learning methods. Readers with an interest however, may want to inspect Appendix C in which the most-prominent inductive learning methods for SWM will briefly be touched upon.

4.1 Data-Mining Tasks

The graph-based RDF model forms one of the pillars onto which all the SW is built. Hence, the more-prominent approaches in the literature focus on learning at this level. The tasks that their learning algorithms strive to complete are classification, prediction, and clustering. These three tasks will be discussed next. Please note that these low-level tasks are generally used to construct more-complex tasks. Several mature examples of such tasks will be discussed in section 4.2.

4.1.1 Classification

Recall that classification involves learning a predictive model that is capable of correctly assigning new instances of unknown classes to one of several predefined classes. With LD, these instances can be either resources or the relations between these resources. If desired, this scheme may be extended to a **multi-label classification** task in which an instance may belong to more than one class, thereby providing a probability for each of the classes (Rettinger, et al. 2012). A multi-label scheme may be useful when reasoning with uncertainty, as instances may not always belong unambiguously to one single class.

With propositional DM, classification is often performed on each instance independently. In the case of LD however, the local neighbourhood of an instance may very well interact with that instance. Therefore, these should be taken into account; a form of classification known as **Collective Classification** (Sen, et al. 2008, Tresp, et al. 2008). One important aspect of collective classification is the propagation of newly classified instances. More specifically, as the class of an instance depends on the distribution of the classes amongst its local neighbourhood, any change in that distribution will influence the class of the instance at hand as well.

4.1.1.1 Subject Classification

Subject Classification concerns assigning a resource – the *subject* – to a specific class of resources (Getoor and Taskar 2007, Rettinger, et al. 2012). Differently put, based on the characteristics of a certain resource it may be possible to predict to which class it most likely belongs (Figure 4-1).

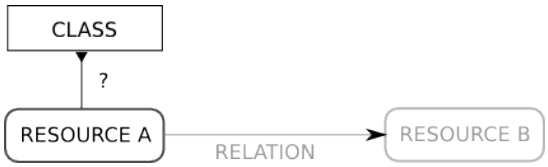

Figure 4-1: Subject Classification concerns predicting the best-fitting class of a resource.

As an example, consider the case when enriching the triple store with a new resource about a recently-found wooden object of an unspecified class. By comparing the characteristics of the new addition with those already in the triple store, a multi-label classifier might assign it to the class of *Artefact* and *Ecofact* with a probability of 0.82 and a 0.18 respectively. If a single class is desired, the one with the highest probability may be selected, which in this case would result in the object being classified as an artefact.

4.1.1.2 Predicate Classification

Predicate Classification concerns assigning a relation – the *predicate* – to a specific class of relations (Getoor and Taskar 2007, Rettinger, et al. 2012). In other words, based on the characteristics of a certain relation it may be possible to predict to which class it most likely belongs (Figure 4-2).

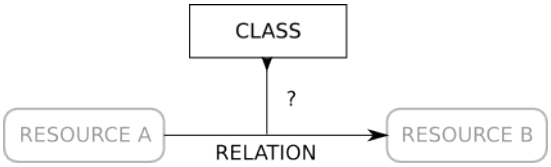

Figure 4-2: Predicate Classification concerns predicting the class that best fits a certain relation.

As an example, consider the case when enriching the triple store with recently discovered knowledge that a certain amphora was used as a container for olive oil. This new fact might be stated by adding the relation *had fluid contents* between the corresponding resource and a resource on olive oil. By comparing of the domain and range of this relation, as well as those of the resources it connects, a simple classifier might determine that the relation *had fluid contents* is an instance of the class *had contents*.

4.1.1.3 Duplication Detection

Duplication Detection concerns determining which resources refer to the same entities (Singla and Domingos 2006, Rettinger, et al. 2012). Differently put, it aims to detect identical resources occurring more than once in the data. Discovering these conflicts is generally wrapped into a classification task, thereby it will have exactly two classes; one for resources that are identical, and one for those that are not. Correctly resolving those classified as identical is an essential step in improving the quality of that data. This final step however, is not necessarily part of a KDD process.

As an example, consider the case when enriching the triple store with another RDF tree. Such integration has the possibility of creating duplicates, especially when the source and the domain of the additions are the same as those already in the store. By applying a binary classification method, each resource in the newly added tree is compared to the already stored resources, thereby limiting this latter set to plausible candidates only. A possible result might be that only one duplicate resource is found. The simplest, although probably undesirable, solution is to delete the newer of the two resources. Alternatively, a merging approach may be applied.

4.1.2 Prediction

A prediction task is a special form of classification that involves determining a value of some sort (Rettinger, et al. 2012). In the case of LD, such a value can either be a resource, a literal, or a relation. Similar to classification, this scheme may be extended to a **multi-label prediction** task in which one or more values are possible, thereby providing a probability for each of them. Such a scheme may be useful when reasoning with uncertainty, as instances may not always fit one distinct value.

Predictions may be made by use of, either by a constrained or unconstrained approach (Rettinger, et al. 2012). In the former case, an ontology limits the range of possible values by actively ensuring that the validity remains intact. The latter case however, does not necessarily perform such consistency checks.

4.1.2.1 Predicate-Object Prediction

Given a certain resource, Predicate-Object Prediction concerns predicting which other resource or literal – the *object* – fits optimally with a relation – the *predicate* – of the original resource (Rettinger, et al. 2012, Getoor and Taskar 2007). That is, provided that we have a relation with a resource on one side, predict the resource on the other side (Figure 4-3). This does not limit the predictions to a subset of the resources, as a large number of *objects* are also *subjects* in some other relation. Note that the exact task would depend on the type of the *object*; while a classification task would most naturally fit with a resource-type or textual-literal *object*, a regression task might suit better with a numeric-literal *object*.

Figure 4-3: Predicate-Object Prediction concerns prediction the resource that fits best to another resource with respect to a certain relation.

As an example, consider the case of wanting to enrich a number of resources on amphorae with respect to how they were used. This might utilize the relation *secondary use*, of which the range specifies the usage beyond that which was originally intended for the domain. Within the context of amphorae, this range might contain prizes, funeral decorations, and grave markers. If a multi-label prediction scheme were to be used, each of these would have a certain probability. These probabilities would have been determined based on the shape, the colour, and the possible remaining contents of the amphora in question, as well as on the archaeological context in which it was discovered.

4.1.2.2 Predicate Prediction

Given two resources, Predicate Prediction concerns predicting the relation – the *predicate* – between these resources (Rettinger, et al. 2012). Therefore, this is also known as relation or link prediction (Figure 4-4). Depending on the set of possible relations, the to-be-connected resources may be either from the same class or from different classes.

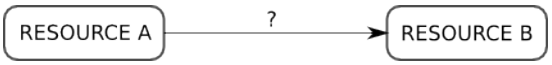

Figure 4-4: Predicate Prediction concerns predicting the best-fitting relation between two resources.

As an example, consider the case when two or more amphorae are found to be made by the same pottery workshop during the same period. That is, a learning method discovers strong similarities between the involved amphorae, thus resulting in a high probability that they are related. Provided that these are not mere duplicates, a relation such as *same origin* can be inferred.

4.1.3 Clustering

The most-common use of clustering algorithms on LD is to automatically generate a taxonomy of a data set (Figure 4-5) (Stumme, Hotho and Berendt 2006, Berendt, et al. 2004, Maedche and Zacharias 2002). Typically, once such a taxonomy has been generated, it will be manually extended to a full-fledged ontology.

Generating a taxonomy from clusters requires the latter to be structured hierarchically. Whereas a standard clustering algorithm forms flat clusters, i.e. all clusters are siblings of each other, dedicated hierarchical-clustering algorithms do exist (Stumme, Hotho and Berendt 2006, Berendt, et al. 2004, Maedche and Zacharias 2002). In their most-basic form however, such algorithms will produce a hierarchy within the data without a (human-interpretable) rationale. Fortunately, by extending this to a

conceptual hierarchical-clustering algorithm, each of the clusters may be provided by a human-interpretable label (Fisher 1987, Fanizzi, d'Amato and Esposito 2008).

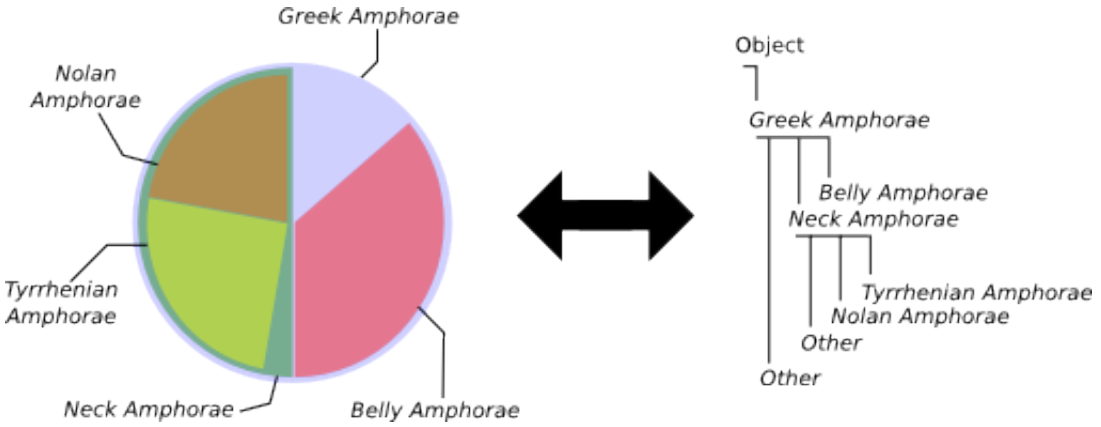

Figure 4-5: Example of generating a taxonomy (right) from an hierarchically-clustered data set (left). Here, assume this is a dataset about Greek amphorae that is clustered based first on civilization, second on shape, and third on location. Note that the labels would generally be more descriptive instead of the ones used here.

Thus far, only the generation of a taxonomy from *new* data has been considered. The same method however, may also be applied to existing data sets that already adhere to a certain ontology. In those cases, an alternative taxonomy may be offered which is based on similarities between the entities instead of on a predefined taxonomy, which was made by domain experts. For instance, instead of finding *Nero* listed under *Roman Emperor*, he might alternatively be found in the cluster containing resources concerning *Roman Cities* and *Disaster*. Due to the latter being based on the data itself, it will be less likely to possess a user-induced bias with respect to the hierarchy. However, the occurrence of erroneous data and variation in the data's quality might limit such effect.

Instead of clustering a whole data set it could alternatively be limited to a specific (combination of) subgraph(s). As such, this approach may be applied to the results returned by a query. An advantage of this is that the clusters and their hierarchy are based on the local neighbourhood of ones' query, thus offering only information on relevant data. In addition, each action that moves, narrows, or broadens the scope of the search will trigger a recalculation of the clusters, thus, once again, providing information only on the most-relevant data. Furthermore, by clustering similar results, the user is presented with a less-exhaustive list, thereby allowing for a more user-friendly and intuitive browsing climate.

4.2 Applicable Solutions

The majority of the developments in SWM are still fairly academic. Even so, there have also been a fair number of projects that have resulted in a more-refined and usable product. Some of these might even be suited for ARIADNE, albeit with a number of adjustments. Therefore, a cross section of the existing solutions will be discussed next. Note that, for readability, most technical details will be omitted.

4.2.1 SPARQL extensions

Recall that SPARQL is the recommended (query-based) interface for a triple store. As a result, nearly all triple stores support this standard. Hence, the integration of DM with SPARQL might provide a clean and natural solution. Two different types of extensions that implement such capabilities will be discussed next.

4.2.1.1 Assisted Query Forming

While SPARQL has long been the recommended standard for querying a triple store, its complexity still forms a barrier for many users. A large part of this complexity originates from the heterogeneity of the data, which may be mapped to very different ontologies. It is unlikely that the average user is familiar with all of these variations, thus resulting in an inability to construct queries that would optimally exploit the data. Therefore, several projects have focussed on extending SPARQL with the ability to assist in the forming of queries (Figure 4-6). Two different approaches will be briefly discussed next.

SPACE is a query-driven autocompletion extension to SPARQL (Kramer, Dividino and Gröner 2013). At its heart lies an index build from past queries fired at endpoints. The rationale for this approach stems from the idea that the query logs of a specific endpoint provide a good representation of the data to which that endpoint provides access. Therefore, while a query is being written, SPACE compares the progressed query to those in its index for the corresponding endpoint and subsequently suggests the most similar past query.

An alternative to a query-driven approach is that of a data-driven approach. (Gombos and Kiss 2014, Campinas, et al. 2012). That is, instead of predicting a query based on previous queries, restrict the possible queries to what the data can offer. One such method consists of generating a graph summary of the RDF graphs. Once built, it represents a generalization of the original graph from which often-paired RDF elements can be queried. For instance, it might link the predicate *written by* to the class *author*. Consequently, if the latest-written term of a query would consist of the predicate *written by*, a suggestion of the class *author*, e.g. *D. Wheatley*, might be given.

```
1  SELECT * WHERE {
2     ?Article akt:hasAuthor ?Author;
3        a akt:Article-Reference;
4        <|
5  }
6          akt:cites-pubblication-reference
7          akt:has-date
8          akt:has-title
9          akt:article-of-journal
10
11
12
```

Figure 4-6: Implemented example of a SPARQL assisted query formulation (Campinas, et al. 2012).

4.2.1.2 SPARQL-ML

With SPARQL-ML, a wide range of prediction and classification methods are added to the SPARQL interface (Kiefer, Bernstein and Locher 2008, Locher 2007). These methods stem from SRL and have been modified to work directly to graph data. In addition, the methods can be accessed through statements that follow the SPARQL grammar and which are similar to those used by Microsoft's Data-Mining Extension[18].

Under the hood, SPARQL-ML requires the data is stored in the MonetDB[19] database, which supports both relational and graph data. In addition, it needs the Weka[20] and Proximity[21] DM APIs, through which all DM operations are processed. These software packages are run on the server side, i.e. on the server that provides a SPARQL-ML interface. Hence, it spares the users the burden of installing additional tools. Furthermore, all normal SPARQL operations remain unaffected.

```
1  SELECT DISTINCT ?person ?award ?prediction ?probability
2  WHERE
3    { ?person ex:hasAward  ?award .
4      ?person ex:hasFriend ?friend .
5      ?friend rdf:type     ?class .
6      ( ?prediction ?probability )
7                sml:mappedPredict ( <http://www.example.org/projectSuccess>
8                                    '?project = ?person' '?success = ?award'
9                                    '?member = ?friend' '?class = ?class' )
10   }
```

Figure 4-7: An example query in SPARQL-ML to learn a predictive model.

[18] Microsoft's Data-Mining Extension, see msdn.microsoft.com/en-us/library/ms132058.aspx
[19] MonetDB, see www.monetdb.org
[20] Weka is an open-source data mining tool for propositional data, see www.cs.waikato.ac.nz/ml/weka/
[21] Proximity is an open-source data mining tool for relational data, see kdl.cs.umass.edu/display/public/Proximity

4.2.2 Ranking Methods

Traditional ranking methods, such as Google's PageRank[22], generally apply a form of an authority-ranking algorithm. These algorithms order pages on the web based on how often they are referred to from authoritative sites. Here, authoritative sites are defined as important and trusted hubs on the web, e.g. due to their influence within the online community.

While traditional ranking methods work quite well on the regular web, they do not contain a mechanism for handling and exploiting semantic relationships. Hence, these are ill suited in the case of the SW. Fortunately, a handful of ranking schemes exist that specifically target the SW. Five of those will briefly be discussed next, thereby omitting those less relevant to ARIADNE.

4.2.2.1 TripleRank

TripleRank is an authority-ranking method that takes the semantics of LD into account (Franz, et al. 2009). This is accomplished by representing the graph as a third-order tensor (C.1.2), which is capable of exploiting these semantics in a natural fashion. By applying a specific factorization method, authoritative sources can be determined. By subsequently calculating the contributions of these sources to a group of triples, an ordering can be found. In addition, groups of semantically-similar predicates and resources may be identified.

4.2.2.2 ReConRank

The ReConRank method is a fusion of two ranking algorithms; ResourceRank and ContextRank. (Hogan, Harth and Decker 2006). The first of these has been adapted from PageRank and thus applies a form of authority ranking. This is accomplished by iteratively going through the graph whilst ignoring the semantics of the connecting links. ContextRank, on the other hand, takes the context graph into account. This graph consists of context-specific resources and predicates which are trusted to be valid. Finally, both ranking algorithms are combined to compute the ReConRank order of relevance.

4.2.2.3 xhRank

xhRank is a ranking approach that endeavours to implement multiple different metrics into one single package in the hope of achieving the best of several worlds (He and Baker 2011). Hereto, it calculates the ranking based on relevance, on importance, and on query length. Of these, the relevance is determined based on the context graph of a query, as well as on the contextual similarity of phrases and terms within that query, and those contained in the RDF graph. The importance is computed by considering authority nodes, as well as the popularity of all relevant resources. Finally, the query length is calculated by evaluating a (weighted) context graph with respect to the input query. Once all metrics have resulted in a (raw) rank, these are combined to form the overall rank.

[22] PageRank, see www.google.com/about/company/products/

4.2.2.4 SemRank

The SemRank relevance model constitutes a fusion of semantic and information theoretic methods, as well as heuristics (Etter and Domeniconi 2014, Anyanwu, Maduko and Sheth 2005). Together, these techniques result in a unified model with which all types of complex semantic relations, known as Semantic Associations (SA) (Figure 4-8), can be ranked by relevance. Instead of applying popular relevance measures, such as shortest path or least-frequently occurring path, SemRank calculates the Information Gain (IG) per relation. This metric conveys how much information a user would gain when presented with the SA to which the IG belongs. However, as the developers acknowledge that different domains require different measures, they provide the option of easily switching to another relevance metric.

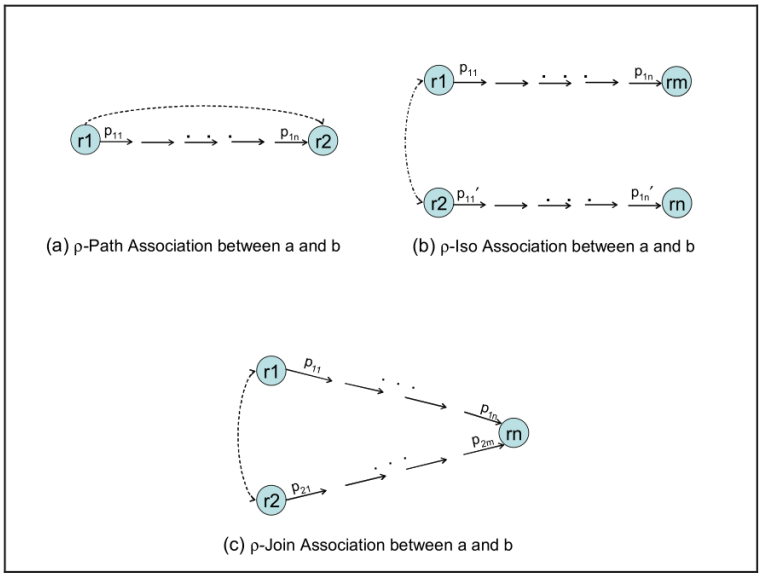

Figure 4-8 : Three types of Semantic Associations (Anyanwu, Maduko and Sheth 2005).

4.2.2.5 Vector Space Models

Vector Space Models (VSM) constitute an approach by which (unstructured) data can easily be searched through to find potentially relevant answers that fit a descriptive query (Appendix B). Such an approach can be added on top of regular SW queries, thereby providing more versatile and more scalable searching abilities. Moreover, as VSMs represent the potential relevance on a continuous scale, these values may be used to rank the corresponding results as well.

Several studies have focused on incorporating VSM into the SW (Freitas, et al. 2012, Mendes, et al. 2011, Castells, Fernandez and Vallet 2007, Tous and Delgado 2006). Hereto, they indexed resources by their vectors. That is, each non-zero term in a vector denoted a predicate-resource pair belonging to that resource. In addition, each of these pairs was weighted to reflect how well they represented their corresponding resource. For instance, given an item in an archaeological data set, the predicate-

resource pair *instance of Dragendorff 33* would be far better at describing the item that the pair *is an artefact*.

The implementation of VSM into the SW can be approached in two different ways. Either a keyword-based query is used to generate a SPARQL query, or a SPARQL query is used from which keywords are extracted. Either way, both query and keyword are, at some point, available for further processing. This process continues by the execution of the query by the query engine, after which the results are ranked based on their similarity to the keywords (Figure 4-9).

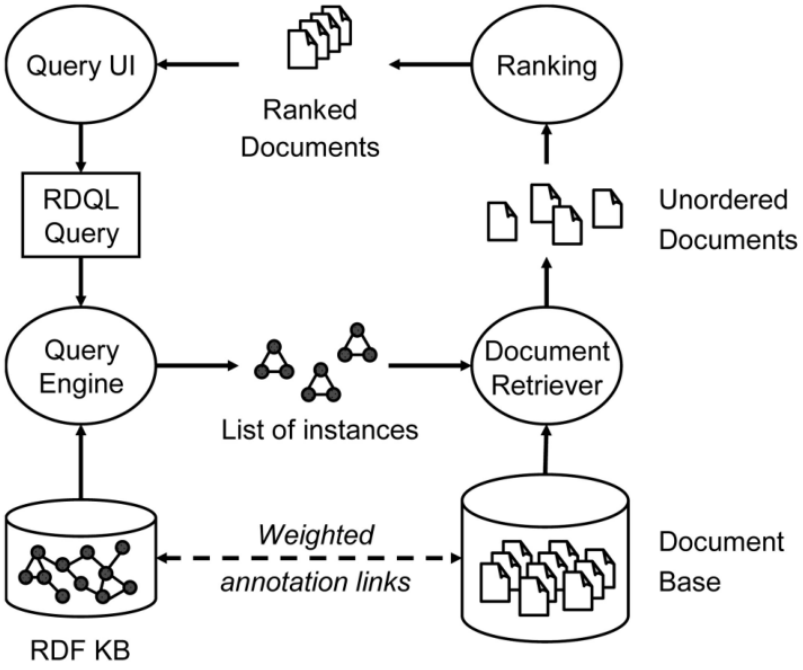

Figure 4-9: Workflow as to how VSM may aid in ranking query results (Castells, Fernandez and Vallet 2007).

4.2.3 Frameworks

Several frameworks exist that focus on adding DM capabilities to graph data or data composed of formal ontologies. Of these, four relevant examples will be discussed next.

4.2.3.1 SUNS

Statistical Unit Node Set (SUNS) is a ML plugin for the Large Knowledge Collider[23] (LarKC); a large-scale integration project aimed at developing a platform for massive distributed incomplete reasoning on the SW. It has since been ported to work on relational data as well (Huang, Tresp and Kriegel, et al. 2009).

[23] Large Knowledge Collider; a FP7 project, see www.larkc.eu

The SUNS framework (Huang, Tresp and Bundschus, et al. 2011, Huang and Tresp 2010) centres on the concepts of *statistical unit* and *population*, which it defines as an instance of a certain class and all instances under consideration, respectively. In addition, it defines each potential triple as a binary *triple node* of which the value is *true* if the triple is known to exist and *false* if the triple is known not to exist. Moreover, the entirety of *triple nodes* that belong to a *statistical unit* is defined as the *statistics unit node set*. For instance, an arbitrary artefact of the class *Dragendorff 33* would be a *statistical unit* of that class. Each triple that states a fact about that artefact, i.e. that it has the colour red, is a positive *triple node*.

At its core, SUNS applies a multivariate[24] prediction algorithm, which is advocated by SUNS' developers as providing an improved predictive performance when compared to traditional algorithms. Regardless, it constitutes a propositional approach, thus requiring the graph data to be translated into a relational matrix. The unknown triples are subsequently predicted by factorization (C.1.1), with the option of integrating this new knowledge into the triple store.

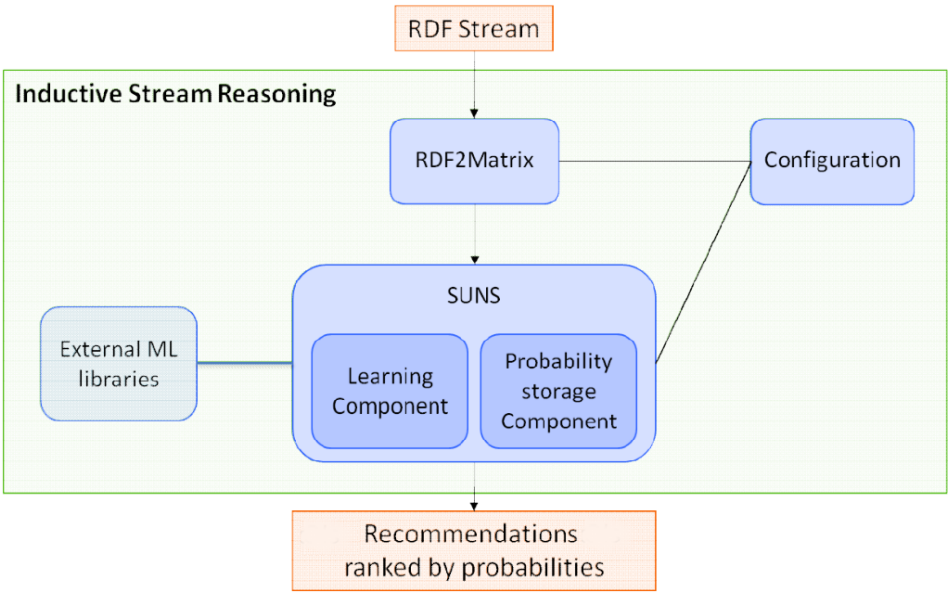

Figure 4-10 : A schematic overview of the SUNS framework (Huang and Tresp 2012).

4.2.3.2 LiDDM

The Linked-Data Data Miner (LiDDM) aims at providing a framework that offers a LD-specific alternative to the more-regular KDD schema (Figure 4-11) (Narasimha, et al. 2011, Ramezani, Saraee and Nematbakhsh 2013). Moreover, it emphasizes the notion of simplicity with the developers advocating the use of regular SPARQL queries in order to prevent alienating the user with a complex learning curve.

[24] Multivariate methods are a fusion of supervised and unsupervised methods, which use known input features to predict several variables jointly, thereby generally increasing their predictive strength.

The first step in the LiDDM framework is the import of data as formulated as a SPARQL query. By going through several pre-processing steps the data is then, among other things, translated to a propositional format. Once completed, propositional DM methods can be applied as provided by an external DM processing engine. The results of this can subsequently be visualized.

In order to test their framework, the developers created the LiDDM Tool (LiDDMT). To facilitate the DM methods they implemented the Weka API. Based on the results gained, they conclude that the strength in the proposed framework lies in mining several aggregated data sets simultaneously, thereby offering a flexibility with respect to the data format. In addition, the developers hope to automate many of its features in the near future.

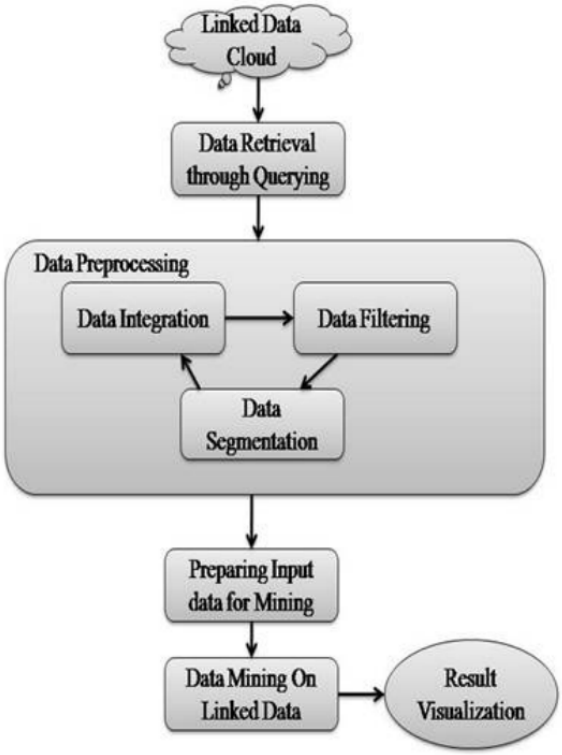

Figure 4-11 : Schematic depiction of the LiDDM architecture (Narasimha, et al. 2011).

4.2.3.3 OLAP

In the field of Business Intelligence, Online Analytical Processing (OLAP) involves a framework for the analysis of multidimensional relational data (Codd, Codd and Salley 1993). Typically, this concerns an interactive DM process from which the results may lead to business and financial reports.

The core concept of any OLAP application is the OLAP Cube (Figure 4-12). Simply put, an OLAP Cube represents a generalization of tabular data, thereby placing certain aspects of a multidimensional data set on the axes of the cube. When requiring more than the three dimensions that a cube provides, it is customary to speak of an OLAP Hypercube. New insights may next be obtained by interactively selecting

and analysing slices of this (hyper) cube. For instance, data on artefacts, location of their discovery, and their carbon dating might be brought together in a three-dimensional OLAP Cube to analyse the possible relations between them.

Analysing data on the SW by OLAP has slowly been gaining momentum. However, the current focus leans more towards the translation of relational data to an OLAP representation on the SW. To accomplish this, the W3C recommends the use of the RDF Data Cube ontology (QB) (Cyganiak and Reynolds 2014). Others however, deem this ontology too limited (Etcheverry and Vaisman 2012, ragimov, et al. 2014), and have extended it with QB4OLAP to enable all analytical abilities of OLAP. As a result, relational data published on the SW using the QB ontology can be analysed with OLAP techniques by using QB4OLAP. A more native alternative is to use the Open Cube vocabulary (Etcheverry and Vaisman 2012), which combines QB and QB4OLAP into a single ontology. Moreover, it allows for performing OLAP operations via SPARQL queries.

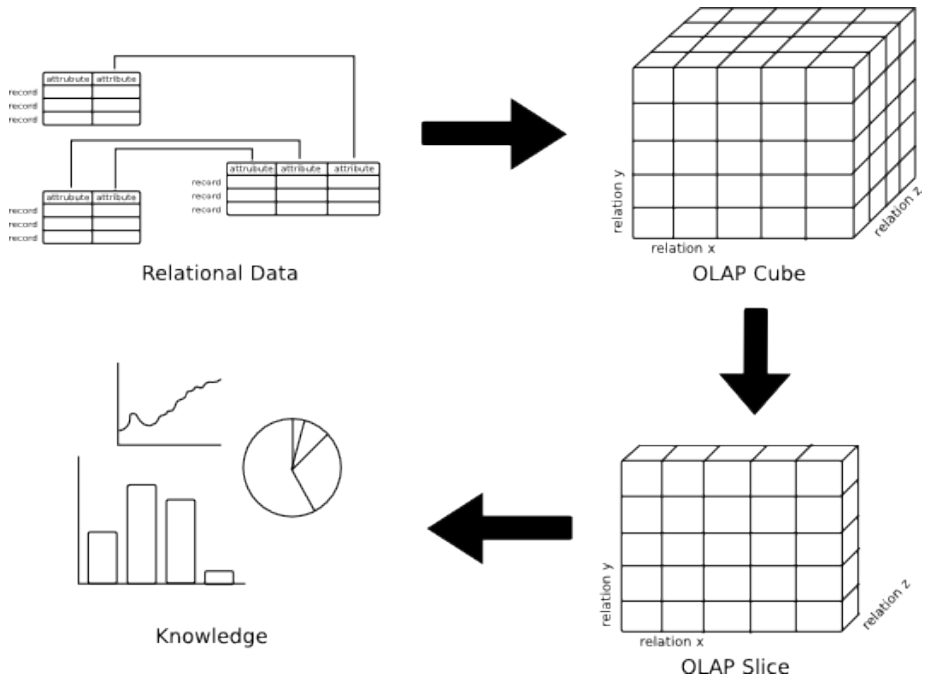

Figure 4-12: Schematic representation of an OLAP workflow.

4.2.3.4 AITION

AITION is an interactive DM solution for the biomedical domain (Dimitropoulos, et al. 2012, Metaxas, et al. 2014). Developed by the University of Athens[25] for the FP6 Health-e-Child[26] project, it specifically aims at discovering knowledge in a medical processing environment. More specific, it provides a full KDD solution by which (biomedical) researchers can pre-process, simulate, and visualize relational data, as

[25] University of Athens, see www.uoa.gr
[26] Health-e Child, see www.health-e-child.org

well as construct statistical models and subsequently infer from them (*Figure 4-13*). Hereto, AITION offers a user-friendly graphical interface similar to those of statistical software packages. This interface allows one to tweak the KDD process to his or her needs, amongst which are the selection of algorithm and the optional specification of prior knowledge and medical ontologies.

Originally, AITION was developed as a stand-alone desktop application. However, due to limitations in processing capabilities it was later extended to a server-oriented design. This allows it to run on distributed architectures such as clusters, grids, and clouds. These architectures however, should provide access to a relational (big data) database for AITION to work properly. Hence, AITION expects the data to conform to the specifications of MRDM. That is, graph-based data such as LD is unsupported at this time.

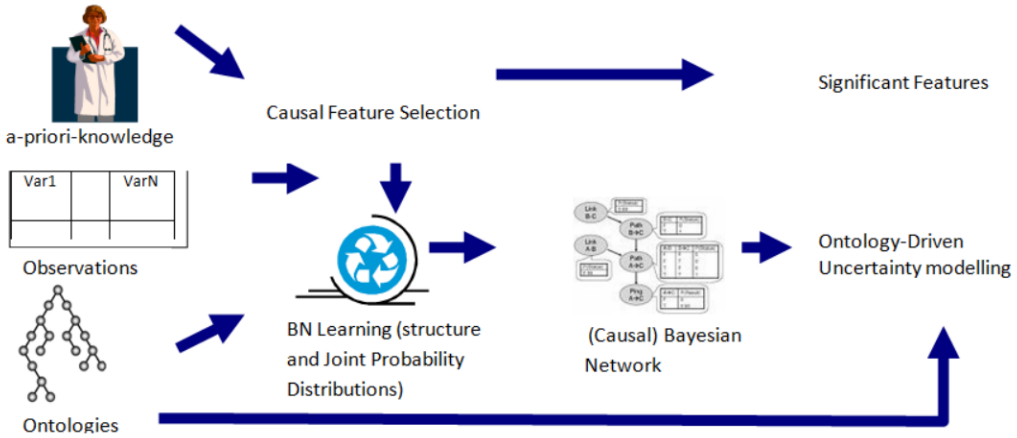

Figure 4-13: Schematic depiction of the AITION framework (Dimitropoulos, et al. 2012).

4.2.4 Platforms

Whereby previous solutions were either mostly theoretical, or extensions to triple stores, the following two platforms constitute relatively complete products with a refined user interface. These will be touched on next.

4.2.4.1 Rapidminer

Rapidminer is a popular DM and Business Analytics platform that aims at providing the whole KDD process to business users. To this end, it combines a wide range of DM and DM-related techniques with an intuitive interface (Figure 4-14). Moreover, its developers try to stay at the front of technological innovation, thereby offering versions capable of running on high-performance and distributed architectures, as well as running directly from the cloud.

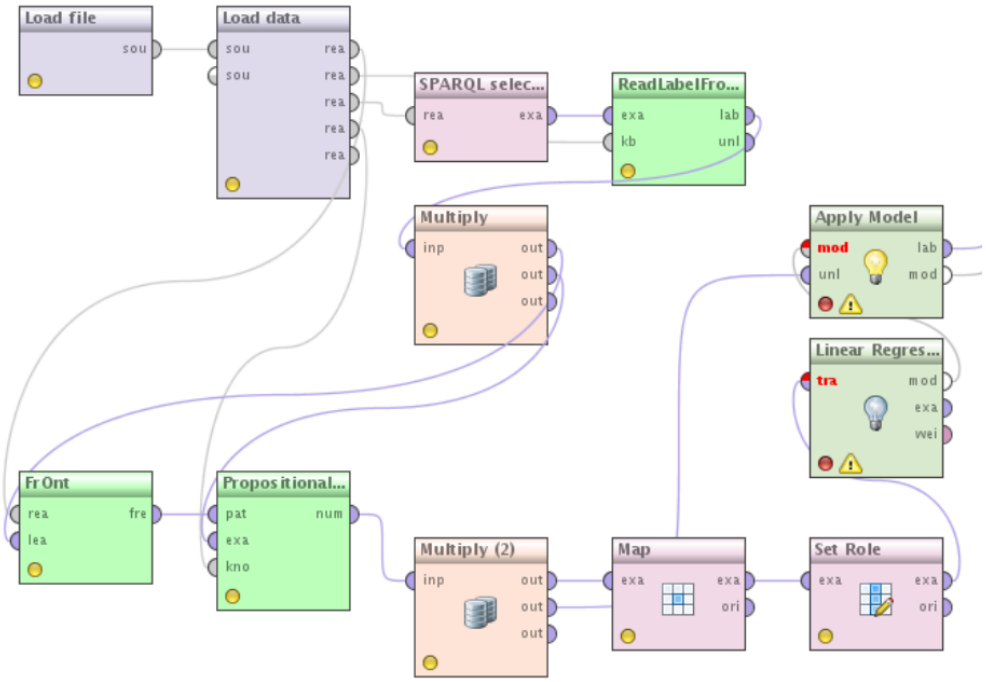

Figure 4-14: Example of how a DM workflow is depicted in Rapidminer (Potoniec and Lawrynowicz 2011).

4.2.4.1.1 Extensions

While Rapidminer already offers a considerable number of possibilities out of the box, its largest strength may be its library of third-party extensions. At the time of writing, three extensions added support for the SW: the Linked Open Data, the SemWeb, and the Rmonto extension.

Linked Open Data (LOD) extension

The LOD extension[27] is the most mature of the three SW extensions (Ristoski, Bizer and Paulheim 2014, Paulheim and Fümkranz 2012), thereby offering both a novice and expert interface. In addition, it allows for the importing of RDF data either locally or remotely through SPARQL queries. Furthermore, it makes it possible to form links between these data sets, as well as following those and other links throughout the SW. By using these capabilities, an automatic selection of relevant features in the data can be made. These features may subsequently be used to discover interesting patterns with. Note however, that this extension emphasizes statistical data.

[27] The Rapidminer LOD extension was previously known as the Weka FeGeLOD extension.

SemWeb extension

The SemWeb extension is a small and still experimental extension that aims to provide pre-processing methods for RDF data that will result in improved learning (Rapidminer SemWeb n.d., Khan, Grimnes and Dengel 2010). At its heart, the extension offers two new data transformation techniques that allow for the comparison of triples by means of several similarity metrics. Once applied, propositional DM methods can be used to explore the data.

Rmonto extension

The Rmonto extension aims to provide DM and ML capabilities to data based on formal ontologies (Potoniec and Lawrynowicz 2011, Potoniec and Lawrynowicz 2011b). Hereto, it offers an internal graph-based representation, thus preventing the need for propositionalization. However, this does cause the majority of Rapidminer's capabilities to be inapplicable. Rmonto provides several adapted versions of these capabilities, all of which are unsupervised learners. According to their roadmap, they endeavour to expand the list of capabilities in the near future.

4.2.4.2 g-SEGS

As a generic variant of the Searching of Enriched Gene Sets (SEGS), a gene discriminator which uses biological ontologies as background knowledge, g-SEGS (generalized SEGS) is a domain-independent semantic DM system built on top of the Orange4WS[28] DM environment (Figure 4-15) (Lavrač, et al. 2011, Novak, et al. 2009). Motivated by the success of SEGS, g-SEGS specifically aims at discovering interesting patterns in data sets that are specified using an OWL ontology.

The g-SEGS system operates by defining the search space from the hierarchy of all *is-a* relations in an ontology. The corresponding data is subsequently translated to a propositional form and used to guide the search for a hypothesis within the data. That is, the search space is being searched and pruned until several interesting patterns are discovered which might explain the data. These patterns are then described by logical rules, thereby allowing for a simple integration into a knowledge base.

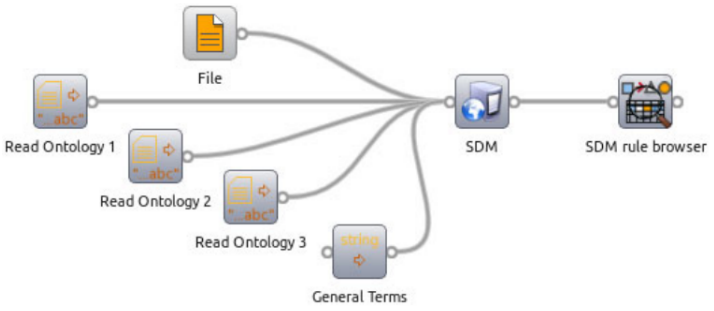

Figure 4-15 : The workflow as depicted by g-SEGS running on Orange4WS environment (Lavrač, et al. 2011).

[28] Orange4WS, see orange4ws.ijs.si

4.2.5 Summary

The following decision criteria should be considered when evaluating the applicability of a solution to ARIADNE (*Table 4-1*):

User-controlled DM process

The majority of the solutions that were discussed provide a user interface (UI) with which the DM process can be controlled. Save for *SPARQL-ML*, all of these featured a seemingly user-friendly graphical UI. In addition, *SPARQL-ML*, *Rapidminer*, and *LiDDM* require a relatively detailed specification of the DM process, thus having a steeper learning curve. *g-SEGS* and *AITION* in turn, lead the user through a number of simple steps, e.g. the adding of a data source. In the case of *OLAP*, it depends on the selected backend.

DM Abilities

Of all solutions, *Rapidminer* is the most versatile with respect to the different DM abilities that may be applied. Somewhat more-limited is *SPARQL-ML*, which provides a subset of all SRL methods. One-track solutions are *SPARQL assist*, *ranking methods*, and *SUNS*, which focus on predicting relevant data. Similarly, *g-SEGS* focusses solely on discovering interesting association rules. In case of the remainder, all of which are frameworks, it all depends on the DM backend.

SW compliant

Due to having their roots in relational databases, both *OLAP* and *AITION* would require quite some work to enable them to operate on the SW. All other discussed solutions allow for the importing of graph data. Save for *g-SEGS* and *SUNS*, all of these mention the capability of importing the data directly through SPARQL. However, only *SPARQL-ML*, *SPARQL Assist*, and *Rapidminer* ensure a native processing of graph data; no propositionalization is required.

Table 4-1 : Overview on the relevant characteristics of the different solutions to DM on LD

	Data Input Type	*Data Input Source*	*DM UI*	*DM Abilities*	*Propositionalization*	*Visualization*
SPARQL Assist	Graph	SPARQL	No	Prediction	No	No
SPARQL-ML	Graph	SPARQL	Yes	Adapted SRL	No	No
Ranking Methods	Graph	SPARQL	No	Prediction	Depends†	No
SUNS	Graph	Unspecified	No	Prediction	Yes	No
LiDDM	Graph	SPARQL	Yes	Depends†	Yes	Yes
OLAP-like	Relational	Database / SPARQL	Yes	Depends†	-	Yes
AITION	Relational	Database / File	Yes	Depends†	-	Yes
Rapidminer	Any‡	Any‡	Yes	Any‡	Any‡	Yes
g-SEGS	Graph	File	Yes	Rule Mining	Yes	Yes

† Depends on which algorithm or DM backend has been selected

‡ Depends on which extension has been selected

ARIADNE's official proposal expresses the desire to allow its users to perform a variety of DM operations by themselves. For this to be possible, a user interface (UI) should facilitate access to these operations. Hence, those solutions that provide a 'DM UI' would be preferred above those that do not. Fortunately, the majority of the listed solutions do offer such capability. In addition, to prevent unnecessary overhead, propositionalization methods should be avoided if possible. These preferences narrow our list of potential options to a mere two solutions however; SPARQL-ML and Rapidminer. Whether either, both, or neither of these two will be suitable for integration within ARIADNE will depend on the wishes of the target users as well as on the data to which their operations will need to apply.

5 Domain Understanding

Understanding the domain to which DM is to be applied is of crucial importance. In fact, it constitutes the first step in any KDD process. By developing that understanding, we expect to get a proper overview of the needs and wishes of ARIADNE's potential users, as well as of the tasks they will likely perform. To accomplish this, we will superimpose a framework dedicated to user-driven tasks within an (digital) information infrastructure (Amin, et al. 2008, Kellar, Watters and Inkpen 2007). These tasks may be classified under three high-level goals, which in no particular order, concern the seeking, exchanging, and maintenance of information (*Figure 5-1*). The corresponding Information Tasks (IT) are as follows.

Fact Finding concerns a goal-oriented and focused search to retrieve a specific piece of information. For instance, consider wanting to know the year in which Emperor Augustus passed away; a straightforward question requiring a simple query.

Information Gathering involves performing several search tasks intended to fulfill a higher-level objective, such as collecting information to form a decision, or with which to test a hypothesis. For example, consider wanting to determine which Roman settlements were vital to local trading in a particular area; a question requiring extensive research through multiple sources of information.

Keeping Up-to-data concerns a search task, generally not goal-driven, to discover what may have recently changed. For instance, consider browsing through the list of recently-added data sets to determine whether any of them may be interesting.

Transactions involve the exchange of information by either retrieving or storing it. As an example, consider the acts of downloading and uploading a data set, respectively.

Communication concerns the exchange of information by face-to-face communication or by technological means such as by e-mail. For instance, consider a strong desire to discuss the Rabbit of Caerbannog by means of a video conference.

Maintenance involves the organization of information, such as updating the content and solving any issues that surface. For example, consider aligning several ontologies used in a triple store. Within ARIADNE, this IT will only be relevant to repository and data managers.

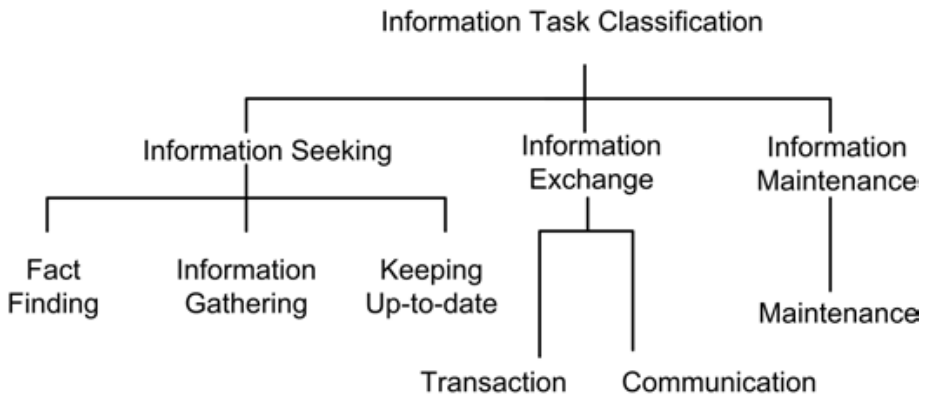

Figure 5-1: Classification of Information Tasks. Adapted variant by (Amin, et al. 2008) *from* (Kellar, Watters and Inkpen 2007).

When relating the ITs with LD, a few observations can be made. First of all, due to its unambiguous properties and its powerful reasoning abilities, LD appears to be a suitable approach to aid in fact finding. In addition, its ability to link and aggregate the information within data sets, as well as promoting a single data standard, provides a large benefit when gathering information. These latter characteristics also contribute to an easier more exchange of that information, as well as providing a less-complex maintenance task.

5.1 Relevant Studies

According to Deliverable 2.1, the ARIADNE stakeholder survey (*n = 744*) amongst 692 researchers and 52 data providers (Selhofer and Geser 2014), the main challenges that confront them during their daily work are those of data transparency, data quality, and data accessibility. Here, transparency is described as having a good overview of the available data, quality as a measure of completeness and structure of that data, and accessibility as the restrictions and high costs involved with acquiring the data (Selhofer and Geser 2014). Note that the latter challenge is out of the scope of DM, and consequently, this report.

Of the reported challenges, only **data transparency** and **data quality** are relevant to consider as areas to which DM may prove beneficial. These challenges roughly correspond to the two ITs of *information seeking* and *information maintenance*. That is, an improvement in data transparency will likely improve the ease at which relevant information can be found. Similarly, an increase in data quality will likely be the result of a good maintenance procedure.

Overall, the researchers appear to be unaware of the potential possibilities and benefits that a DM solution could offer (Selhofer and Geser 2014). A large number of them are even unaware as to what the term *Data Mining* actually means, and express various wishes of which most will already be fulfilled by LD alone. Therefore, these users provide little insight into the general expectations of the target audience with respect to DM. Unfortunately, save for a few individuals– this issue appears to extend to nearly the entire sample group that participated in the survey. A large contributor to this might be the

apparent lack of currently-available DM applications within the domain. That is, of the users that participated in the survey, less than a fifth rated their availability good (12%) or very good (5%). This state of affairs prevents users from gaining any practical experience, and thus impeding them in forming a realistic view of the added value that DM could offer.

In the case of data enhancement, only few stakeholders of the sample group – all affiliated with data repositories – appear to hold the belief that DM might be beneficial. In fact, only one of them strongly beliefs that the integration of DM in repositories should be regarded as a very important aspect. Additional interviews (*n = 3*) were conducted by (Hollander and Hoogerwerf 2014) as part of ARIADNE Deliverable 13.1, from which a desire to *enhance metadata* was concluded. This enhancement would entail, among other thing, automated metadata extraction, duplication detection, link prediction, and overall semantic enrichment.

During the survey, the sample group appeared to hold mixed beliefs about whether DM is an important aspect of their research (Selhofer and Geser 2014). More specifically, half of them deemed it either very important (16%) or rather important (34%), while the remainder thought it to be rather unimportant (26%) or very unimportant (23%). Even less of the survey's participants had ever used a DM solution during their research, with less than a tenth (8%) having used it very often, and slightly more (16%) having applied it frequently. If fact, near half (41%) had (almost) never used DM at all. These numbers may partially be inaccurate given that, as indicated earlier, many participants were unaware of the possibilities that the integration of DM tools into their research may provide, and may, in fact, have used such tools unwittingly. Examples of such tools are those integrated into a GIS, with which advanced analysis of archaeological data is already possible (Conolly and Lake 2006).

According to the survey's participants, the most important type of data they are using during their research is, unsurprisingly, that of excavation data (Selhofer and Geser 2014). Of the other important sources, many appear to involve a geographical component, such as GIS or satellite data. Therefore, augmenting those types of data with the help of DM would likely offer the most effective improvement during the knowledge-discovery stage as perceived by the user. However, it may also be true that researchers are using the other types of data less often due to the poor transparency that currently exists. Irrespective, Hollander and Hoogerwerf (2014) emphasize the need to present the user with, among others, a geo-integrated search, thereby offering numerous options such as an interactive timeline and various background layers.

Another important part of data transparency is the ability to easily distil relevant information from exhausting quantities of *potentially-relevant* information. One approach to achieve this might be to apply some form of ranking to the data (Hollander and Hoogerwerf 2014), either automatically or manually by the user. Alternatively, a combination of the two may be possible, whereby the users can correct or otherwise influence the automatically generated rankings.

5.2 Wishes of Domain Experts

In an effort to gain additional insight into the needs of archaeological researchers we organized five one-hour brainstorming sessions. Four of these were held at Leiden University and one at the VU University Amsterdam. Each of these sessions consisted of an open discussion with a different archaeological specialist, among which were junior and senior researchers as well as Ph.D. candidates. Furthermore, their fields of expertise ranged from Prehistoric hunter gatherers and farming communities, to long-term developments of settlements, land-use dynamics, and spatial dynamic modelling.

The participants were asked to write down several archaeological research scenarios, which they deemed difficult with the current information infrastructure. Moreover, the assumption was made that all archaeological publications and data would have been made available in one large database. In total, the sessions resulted in 26 scenarios, of which an English translation is given in Appendix D.

An analysis of the scenarios posed revealed that the large majority (77%) of the difficulties that researchers experience concern an information-gathering task. Hence, the aggregation of data is desired. In several occasions, these data are likely to originate from multiple, distinct sources and stem from different domains, amongst which are plant and dental records. Moreover, nearly half of these information-gathering scenarios (40%) relate some archaeological entity to a geospatial component, thereby making this latter domain the second most prominent. Finally, three scenarios mention difficulty in finding relevant data using a keyword-based search engine.

The remaining research scenarios (23%) involve a fact-finding task, among which are the search for contact information, specific GIS files, and the number of a physical storage container. Two scenarios that stand out concern the trustworthiness of certain data, which is dependent on the corresponding archaeological context.

Information-Seeking Task	*Number of Scenarios*
Fact Finding	6
Information Gathering	20
Keeping Up-to-date	0

Table 5-1: Dissemination of the archaeological research scenarios into the three distinct Information-Seeking tasks.

When assuming an ideal situation in which all required data has been made available as LD, most, if not all of these information-gathering tasks would be accomplishable without much trouble. In contrast, none of the fact-finding tasks would require LD; for these, a simple relational database would suffice. These results are similar to those of the survey and the interviews. That is, the abilities that are provided by LD are likely to greatly improve the knowledge-discovery process of an archaeological researcher. Also similar, unfortunately, is the difficulty of distilling tasks to which DM might prove beneficial.

5.3 Summary

Based on the previously-discussed survey and interviews, the following points can be distilled which are deemed relevant with respect to DM. Please note however, that a solution to the last point would more likely be considered an issue treatable by ML rather than DM.

Unawareness of the available data reflects the lack of knowledge users possess on what data is available to them. This hinders them during the early stages of their research, as they fail to explore data outside the scope of their current search.

Uncertainty on how to locate relevant data concerns the difficulty of users to get an understanding of the actions required to find and access the data they are looking for. Instead, they run into ambiguity issues with the terms they use to search.

Inability to effectively distil relevant data involves the issues that users experience when faced with an exhaustive list of *potentially relevant* data. To determine what data is relevant and what is not, they need to thoroughly examine each entry in the list.

Incompleteness of Data concerns the perceived gaps in information, due to having been omitted during either the study or the subsequent digitization. As researchers cannot ascertain whether the missing data is of importance, this hinders them from trusting the data set as a whole.

6 Data Understanding

Data Understanding is an important step within any KDD and concerns inspecting the data, their quality, and their abnormalities. Obtaining a good overview of these aspects contributes to the performance of any future DM applications within the ARIADNE infrastructure. After completion of this project at the end of February 2017, it is hoped that Europe's archaeological communities will adopt it. Until that moment however, the available data will be comprised of what already exists, as well as small amounts of new data which will be produced by those involved with Natural Language Processing. Therefore, the conclusions at the end of this report will apply *only* to these data. It is expected however, that these data will constitute a sufficient representation of the future data, and thus allow for generalization, such that the conclusions will hold.

We expect that the large majority of the data that ARIADNE will offer during its first couple of years will be provided by the currently existing digitally-accessible data infrastructures. Only a few of those have already explored the possibility of publishing their data as LD, and even fewer have embraced it entirely. A good example of a repository that *does* embrace LD is the ADS; the Archaeological Data Service based at the University of York, and which has adopted all facets of the LD paradigm for a section of their archive data, and for all their resource discovery metadata. An example of a repository that is well on its way might be EASY[29]; the digital warehouse on Digital Humanities hosted by DANS[30], which offers unstructured data together with their corresponding LD counterpart. In some cases however, data sets consisting solely of LD are published[31].

6.1 Data Produced using Natural-Language Processing

In addition to the Linked Archaeological Data (LAD) from existing infrastructures, ARIADNE will provide LAD that has been generated semi-automatically from unpublished archaeological reports by means of Natural-Language Processing (NLP). These reports, the so-called *grey literature*, are increasingly 'born digital'. Those that are not are scanned to create a copy, which is usually made available in PDF format. In either case, this may be followed by processing the text through specialised tools, to convert the report to LD.

Within ARIADNE, the task of exploring the applicability of NLP falls under WP16 with task number 16.2. The main contributors to this task are the University of South Wales, the ADS, and Leiden University. As it may be assumed that other NLP endeavours will proceed in a similar fashion, there will thus be no

[29] Electronic Archiving System, see easy.dans.knaw.nl/
[30] Data Archiving and Networked Services, see dans.knaw.nl/en
[31] For example, consider the recent CLARIN Dutch Ships and Sailors data set

difference between data from either origin with respect to accessing and retrieving it. That is, both will be stored on the same (or similar) triple store using the same (or similar) ontologies.

The main challenge with which the field of NLP struggles is very similar to that which the Semantic Web tries to solve, namely the problem of making knowledge interpretable by software agents (Briscoe 1991). However, the Semantic Web tries to attack this problem by structuring the knowledge, whereas the field of NLP generally tries to create semantically-aware agents. Unfortunately, due to its complexity, this latter approach is still far from perfect. Therefore, any knowledge engineer should be wary of possible flaws in the data converted by NLP, which were not detected during manual or (semi-) automatic checks.

6.2 Case Study on Data Repositories

Of the existing LAD infrastructures (Fentress 2014), four have been chosen from which the RDF data was deemed a suitable representative of their specific area of expertise. In addition, this selection favoured minimal overlap between the corresponding data sets, as to provide as high a degree of variability as possible. These four are the ADS, EASY, Open Context, and Pleiades.

6.2.1 ADS

Recall that the Archaeological Data Service (ADS) provides a web portal to a triple store. At the moment of writing, it holds the aggregated data of nearly 500 distinct collections. Together, these collections provide 472,172 triples on 103,148 resources[32].

The data appears to be structured in three sections, each with a different purpose. The smallest of these, comprising less than a few percent of the whole graph, describes several types of amphorae by the distinct physical features that make up their shape. The corresponding statements consist largely out of SKOS statements.

With a slightly higher percentage follow numerous descriptions of the sources and studies from which the data originated, all of which mainly make use of Dublin Core (DC). It should be noted however, that most of the DC triples have literals as their object. Furthermore, it appears as if the range of certain DC attributes, e.g. *dc:coverage*, covers disjointed sets such as the disjoint union of temporal period and geographical location.

The large majority of the graph describes the artefacts provided by the sources and studies. The corresponding statements are specified using the CIDOC CRM-EH ontology, of which the properties start with a unique identifier. In addition, descriptions of the same resource frequently occur more than once, with each occurrence specifying a different set of properties.

[32] These figures are based on the graphs retrieved in December 2014.

6.2.2 EASY

Recall that EASY concerns a digital repository on Digital Humanities hosted by DANS, and which experiments with offering LD alongside their corresponding unstructured counterpart, among which are databases, images, and reports. On the whole, the data within EASY's test triple store constitutes more than 25,000 data sets which, together, results in roughly 28,000 resources consisting of 836,447 triples in total[33].

Little variation between concept descriptions exist within the graph, with almost every one consisting of a single data sets with similar properties. These properties mostly concern metadata and are described in Dublin Core[34] (DC) with literal values. Therefore, while the graph is relatively uniformly divided, it does contain little cross-linking.

While not serialised in RDF, EASY additionally provides each data set with more contextual information in the form of a XML file; the "pakbon", or packing slip in English. This file consists of numerous fixed attributes, developed in cooperation with the SIKB[35], that are to be supplied by the submitter of the data set. If converted to RDF, the information herein might prove quite beneficial to archaeological researchers.

6.2.3 Open Context

Open Context[36] is a web portal that allows researchers to publish and access scientific data from various domains, such as zooarchaeological and spatial archaeology, as well as numismatics. At the time of writing, the portal offered access to data from 20 projects, with 34 others forthcoming[37]. Of the corresponding data sets, only a fragment have been converted to RDF. Instead, most data resides in a combination of JSON[38], KML[39], and ArchaeoML[40]. Here, the latter two are XML formats, which aim at expressing geospatial and archaeological data, respectively. However, OpenContext is currently working on aligning its metadata to the CIDOC-CRM.

The currently available RDF data consists of nearly 5,000 triples which, together, describe roughly 1,250 resources[41]. Of these resources, the majority consist of coin, region, and site descriptions. Little variation between concept descriptions exist within or even between these domains, which all apply a static

[33] These figures are based on the graphs retrieved in December 2014.
[34] Dublin Core, see www.dublincore.org
[35] Stichting Infrastructuur Kwaliteitsborging Bodemonderzoek, see http://www.sikb.nl
[36] Open Context, see www.opencontext.org
[37] These figures are based on the graphs retrieved in December 2014.
[38] JavaScript Object Notification, see www.json.org
[39] Keyhole Markup Language (KML), see www.opengeospatial.org/standards/kml/
[40] Archaeo Markup Language (ArchaeoML), see www.opencontext.org/about/concepts/
[41] These figures are based on the graphs retrieved in December 2014.

generic set of properties whereby only the values differ. These values often involve custom resources, as well as those provided by Geonames [42].

6.2.4 Pleiades

Pleiades[43] is a web portal that provides historical geographical information about place from the ancient world. Hereto, they use their own definition of *place*, which constitutes a geographical location with an ancient name, which may vary over time. Currently, the database contains close to 3,500 places, resulting in 2,258,807 triples[44]. Furthermore, the corresponding resources on authors, place types, and time periods consist of an additional 5,000 triples.

Within the triple store each resource type has its own graph, with the place descriptions encompassing nine graphs. These latter graphs constitute the majority of the data. While moderate variation in concept descriptions exist between the type-specific graphs, there is little variation within them.

Particularly interesting is the choice to include an errata graph. The goal of this graph is to hold the falsified statements from other graphs instead of correcting the errors directly. Therefore, both the correct and the incorrect version are available.

6.3 Summary

Based on the case study of four different data sets, as well as on the differences between them, the following notes can be distilled which might require special attention when designing a DM solution.

> **Differences in ontologies used** exist between the data sets from *different* sources, whereas this occurs less so within those from the *same* source. Additionally, the usage of different versions of the same ontology can be observed, which may cause such an ontology to be either under or overrepresented if not dealt with accordingly. Moreover, attention should be given to the use of custom ontologies. Note that, ideally, all data will be translated using a single ontology such as CIDOC CRMarchaeo, CRM-EH, or the ACDM.
>
> **Structural variation** within data sets from the same source appears relatively little, with most of a dataset's concept descriptions following roughly the same structural schema. More specifically, nearly all descriptions in a data set use roughly the same set of properties, which are specified using the same ontologies. The inverse appears true between the data sets from different sources, which all have their own distinct schema structure.

[42] Geonames, see www.geonames.org
[43] Pleiades, see pleiades.stoa.org/
[44] These figures are based on the graphs retrieved in December 2014.

Structurally-flat graphs appear to be quite common. This phenomena occurs due to the scarceness of URIs per resource, either because a resource only has a small number of properties, or because of an extensive use of literals. In the latter case, numerous occurrences of unnecessary use have been observed. That is, literals were used to denote property values for which URIs were available.

Strong Dependency on descriptive values tends to occur frequently throughout all data sets. Often, these descriptions provide crucial information and thus cannot be ignored without a significant loss of knowledge. Therefore, such values should be given additional thought during the development of a DM solution.

Concurrent RDF statements were observed within one data set in which both the correct and the corresponding falsified statements on the same concept were kept available. Due to this construction, duplicate and possibly conflicting entries might surface. Hence, methods for conflict resolution should be explored.

7 Data Mining on Linked Archaeological Data

As discussed earlier, a typical KDD process starts off with a deep understanding of the domain and the data. At this moment, expertise within the former is readily available within ARIADNE. Unfortunately, the same cannot be said about the data aspect. This is understandable, as ARIADNE has reached half of its four-year run. However, as the exploration of data is an important step in any KDD process, this means it is difficult to predict what kind of new knowledge may be brought to light as a result of this process. Therefore, this section will focus on the more-generic options expected to function properly on any form of Linked Archaeological Data (LAD).

Based on both the Domain and Data Understanding step within our selected KDD process, several DM solutions were selected which we deem suitable for a LAD framework such as ARIADNE will likely be. We will next discuss these solutions in more depth.

7.1 Hypothesis Generation

DM methods are capable of detecting patterns in data. Interesting and potentially relevant subsets of these patterns can then be presented to users as starting points for forming new research hypotheses. For instance, the system might detect that specific types of pottery are most often found near coastal areas. This might already be known to the researcher, or it might be something the researcher is interested in exploring further. The interestingness of patterns will be derived algorithmically on the basis of predefined criteria and user feedback. To facilitate this, any LAD repository should ideally offer a KDD interface that provides its users with such capabilities. These capabilities would then be applied directly to the data from one or more repositories.

The integration of KDD capabilities into a structured query interface such as SPARQL or similar would provide a solution that largely satisfies the earlier-mentioned criteria. As a result, any query can easily be extended with data-mining operations capable of generating potential hypotheses. Moreover, as using these additional operations is purely optional, their presence would not hinder users who are solely interested in regular queries. Furthermore, it might prove useful to assist with the formulation of queries for those users who are unfamiliar with the syntax of a structured query language such as SPARQL. Alternatively, the capabilities or several higher-level abstractions could be integrated into a graphical UI, thus lowering their learning curve. Moreover, multiple hypotheses could then easily be presented to users in non-intrusive ways to allow for quick scanning for potentially valuable directions.

7.2 Assisted Query Formulation

The complexity of accessing DM capabilities through a query language might pose a hurdle to interested users, i.e. those who want to use these capabilities but who are unfamiliar with their syntax. To lower this barrier, it might prove useful to assist the user in its formulation of queries. Instead of limiting this assistance to solely the DM capabilities, it may be extended to aid with formulating regular queries as well. This may manifest itself as either predicting or autocompleting the query. Alternatively, a combination of the two may be applied.

Predicting a query involves learning from past queries. Based on a partially-written query, the remainder is predicted by comparing the similarity between the written part and (parts of) past queries. A distinction herein is whether a more local or more global view should be maintained. Here, a more local view would concern queries from users who share a similar background or interest, as well as users who have accessed much of the same data. This however, would require user profiles. In contrast, a global view would consider the queries of all users. A combination is possible as well, thereby favouring more local queries over global ones.

Whereas the previously discussed approach is query-driven, a data-driven approach can be used as well. This would entail learning from the available data, thereby determining frequently occurring combinations of relations which may be offered as suggestions. This could additionally be coupled with relevant ontologies and knowledge of the query language and its DM extension. Such a coupling would prevent offering invalid suggestions as well. Hence, it could be considered as a form of autocompletion.

7.3 Ranking of Query Results

Integrating KDD capabilities into a query interface allows the ability to rank the results of such a query. This ability concerns the ordering of results based upon certain criteria. Within a LAD, a strongly desired criterion is that of relevancy. This criterion is commonly regarded as the most challenging to determine, as it is intertwined with the researchers flow of thought (Franz, et al. 2009). However, when using a query language capable of representing structure and semantics, e.g. SPARQL, determining this relevancy becomes more manageable.

While SPARQL or a similar query language would always form the bridge between a LAD's frontend and backend, it might not necessarily be the interface that is used to search by the researchers. That is, the frontend might facilitate faceted searching, thereby effectively posing as a wrapper to the underlying query language with its DM extension. In addition, it may provide a keyword-based search as well, thereby requiring a NLP solution which translates the query to a SPARQL-based language or similar. While such a translation would likely result in some loss in precision, it does allow for a more user-friendly search. Moreover, it opens up the possibility of incorporating VSM, thereby enabling the traditional ranking of documents based on their relevance to the provided keywords. Note that the

added benefit stems from applying this scheme only to those subgraphs that match the translated query.

Instead of relying on traditional ranking schemes, those specifically designed for the SW can be used. While several innovative and promising algorithms exist, it might be prudent to begin by extending graph-based authority algorithms with the ability to include semantics and structure (Balmin, Hristidis and Papakonstantin 2004, Franz, et al. 2009). Furthermore, graph attributes can be considered as well, among which are (weighted) path length between related resources and (recursively) linked resources (Rocha, Schwabe and Aragao 2004).

An addition to, or as an alternative ranking of individual resources, triples, or documents, is to rank groups of them. More specific, after using the integrate DM engine to cluster any of the above elements based on some criteria, e.g. similarity, the resulting clusters themselves can be ranked. This might facilitate a search method similar to faceted search, thereby letting each of the cluster's boundaries represent a distinct (conjunction of) filter(s). These filters would provide a more natural fit for the data, as well as always being based on the most-recent version. In contrast, those filters used in a faceted search were likely specified by experts prior to implementation, and with only a partial view of the data.

7.4 Resource Recommender System

The resource recommender system concerns predicting resources that might be relevant to the user (Szomszor, et al. 2007). Determining whether an arbitrary resource is relevant may be based on several measures. First, similar to ranking, a distinction should be made as to whether a local or global view should be considered. A combination is possible as well, thereby favouring local recommendations over global ones. Alternatively, the local neighbourhood could be considered instead. For instance, two researchers who separately search through data on Neolithic flint axes with an amber core are likely to be interested in at least some of the resources they access. These resources may thus be suggested to the other researcher, in the form of a recommendation.

A second distinction in recommender systems involves the length of time that is used on which to base recommendations. In the simplest case, only the history of the user's current search will be considered. Typically however, this would result in poor recommendations, as the limited time prevents the creation of an accurate model of the user's interests. For instance, if a researcher searches for Palaeolithic flints on one day, and for Neolithic flints on another, then any recommendation would likely be based only on the most recent history. A more-effective solution is to use the entire history of the user's interaction with the data, provided the LAD's framework at hand will employ some form of user profile (Middleton, Shadbolt and De Roure 2004).

A final consideration concerns which elements of the user's history will be considered when building their profile. That is, which of the recorded elements best represents the interests of a user. Two possibilities might be past queries and accessed resources. A more interesting option is to use the paths

that were traversed through the RDF graph. However, this would entail the storing of user information on a server, which might raise privacy concerns.

Recommender systems generally employ a form of dependency modelling (Hagood 2012, Witten, Frank and Hall 2011). This typically entails the discovery of (weighted) rules that describe the dependencies between resources. A special type thereof are sequence-based dependencies (Berendt, Hotho and Stumme 2002, Stumme, Hotho and Berendt 2005, Stumme, Hotho and Berendt 2006), which additionally take the order of accessing a resource into account. Additionally, the graph structure itself may be exploited, e.g. by favouring recommendations that are closely related to the current or recent topic of interest.

7.5 Data Quality Analysis

Two aspects that reflect poorly on the quality of data are the occurrence of gaps and errors in the knowledge contained therein. To improve these aspects by hand would be a rather time-consuming task. Instead, it would be more feasible to automatically discover these faults, and perhaps even be provided with a possible remedy. Similar to hypothesis generation, this can be thought of as a form of pattern detection. Hence, this constitutes a problem to which DM may provide an answer.

Within the SW, gaps in knowledge manifest themselves as missing resources as well as missing links between them. In addition, resources may miss links to one or more literals. In that case however, the literals are typically missing as well. Filling these voids thus involves predicting the most likely resource, link, or literal (4.1.2). These predictions will hold a certain likelihood of being correct which, ideally, will be closer to *very likely* than to *very unlikely*. As this constitutes a metric with a continuous range it might be prudent to determine a point within this range under which predictions will be discarded. Alternatively, all possible options may be presented to the users, who may determine the validity and relevance of these predictions themselves.

As with all forms of data, the SW is not impervious to the introduction of errors. These errors generally consist of invalid links, unsuitable resources, or erroneous literals. In terms of DM, a large number of these errors involve anomalies within the data, which cannot be explained by any of the discovered patterns alone. These elements may easily be removed depending on the likelihood of them being erroneous. However, it might be safer to present this choice to the users. An additional possibility may be to provide these users with an estimated guess of the correct link or literal. As before, this would require forming predictions. In fact, the results of a removal could even be treated as a new gap in knowledge, thereby allowing for an efficient reuse of the aforementioned prediction method.

As with the generation of hypotheses, an inductive analysis of the data's quality relies on the discovery of relevant patterns. Therefore, the same or similar underlying KDD solution can be applied as well. Such a solution might consist of integrating KDD capabilities into a structured query interface such as SPARQL or similar. This would allow any query to be extended with data-mining operations capable of

discovering erroneous data as well as predicting values for missing data. These capabilities could alternatively be integrated into a graphical UI, thereby increasing their usability.

7.6 Trust Analysis

A large number of archaeologists find it difficult to know which sources of information can truly be trusted; an aspect of paramount importance for using the information in one's own research. Moreover, this trust makes it possible for the researcher to distinguish data that is high quality from that which is not (Artz and Gil 2007). Making such a distinction will become ever more difficult due to the large amount of data that will become accessible through the ARIADNE web portal. Therefore, some suggest treating unchecked **statements as claims rather than as facts** (Bizer and Oldakowski 2004)**.**

In order to determine which data is deemed trustworthy and which is not, a proper representation of trust is needed. For data on the SW (Golbeck, Parsia and Hendler 2003) devised the following nine levels of trustworthiness :

1. Distrust Absolutely
2. Distrust Highly
3. Distrust Moderately
4. Distrust Slightly
5. Neutral
6. Trust Slightly
7. Trust Moderately
8. Trust Highly
9. Trust Absolutely

Initially, each statement would be considered neutral. Based on some measure of trust, this Trust Level (TL) would then either remain unchanged or change to a higher or lower TL. A researcher would be able to see a statement's TL, as well as its change over time. Within ARIADNE, a similar scheme could be used, thus assigning each data set with a TL.

Numerous measures of trust are available. The three most-prominent of which are reputation-based, context-based, and content-based (Bizer and Oldakowski 2004):

> **Reputation-Based Trust** involves rating systems such as used on eBay and Amazon. For it to work, it requires frequent interaction by the system's users, who keep updating the ratings. The drawback is that such a system requires explicit and topic-specific trust ratings, of which the validity strongly depends on the amount of input users provide. Nonetheless, most trust architectures for the SW rely on this approach.

Context-Based Trust considers the metadata on the forming of the data. That is, how was the metadata derived. To this end, it investigates the role of the researchers, including the groups or institutes with which they are affiliated. For instance, one might trust a professor of a widely-acclaimed archaeological institute with direct expertise, more than a student who is new to the domain.

Content-Based Trust applies rules and axioms together with the data's content and related data from other sources. An example of a rule might be that at least five different sources should report the same.

In the case of a LAD repository, all three trust mechanisms could be considered. Here, reputation-based trust would be the most proven and straightforward of the three. However, it would rely on archaeological researchers to rate the studies of colleagues, thereby likely to introduce a bias given the relatively small field. Context-based trust, on the other hand, would allow for the incorporation of metadata within the study, which might include the archaeological context. Therefore, such a mechanism might provide a much more accurate estimation of the true trustworthiness of a study. However, it would strongly depend on expert knowledge, making its development the most resource expensive of the three. The third option of content-based trust could be used to implement more generic and heuristic rules which should work regardless of the data's properties. Finally, as none of the three mechanisms appears to influence the others negatively, a combination of the three might also be an option.

Several studies have focussed on implementing trust mechanisms by means of DM, few of which have tackled the SW. Irrespective, Gupta, Sun and Han (2011) studied the applicability of clustering on trust analysis, thereby grouping data providers based on their TL. The authors report that their methods performed better than traditional approaches to trust analysis, and that interesting clusters were found. Another study performed by Huang, *et al.* (2012), explored methods from SRL with the emphasis on Probabilistic Soft Logic (PSL). Their findings show that PSL might be particularly well suited for trust analysis, and that its performance is similar to traditional techniques. As a final example, consider Taranto, Di Mauro and Esposito (2013), who advocate the notion that trust analysis is strongly related to a specific form of *link prediction*. In addition, they show that a Probabilistic Graph framework is capable of predicting such relations between entities within a graph.

8 Conclusions

This report examined the applicability and feasibility of integrating data mining solutions into ARIADNE; a digital framework to aggregate and integrate archaeological data. Throughout this report, we made the assumption that this data will adhere, either fully or partially, to the principles of the Linked Data paradigm. On the basis of this assumption, we explored various state-of-the-art theories, methods, and solutions to detect patterns in, and establish relations between, linked data from the archaeological domain. Additionally, our study focussed on usage-pattern analysis and content linking, as well as on information retrieval. To this end, a thorough analysis of user's needs and wishes was conducted, as well as an examination of recent and relevant literature and experience of the topics involved.

8.1 Domain Understanding

A good understanding of the archaeological domain is paramount when deciding which data mining tasks may prove useful to ARIADNE's users. Therefore, we conducted an analysis of the questionnaires and interviews that were conducted by WP 2.1 and 13.1, respectively. While providing valuable insight into the stakeholders of ARIADNE, both WPs only touched on the possibility of data mining. Moreover, the large majority of the stakeholders had little to no experience with data mining, and were unaware as to what data mining actually entailed. As a result, very little could be ascertained as to what path any data mining solution should follow. To mitigate this lack of direction, several interviews were held to explore the use of data mining more-actively. Regardless, of all the topics discussed, only a few were relevant with respect to data mining.

In its entirety, the aforementioned studies seemed to indicate that the large majority of the difficulties experienced by stakeholders involve the aggregation of data from different sources, as well as the limitations of a keyword-based search engine. Hence, it appears that Linked Data alone would already provide a revolutionary improvement. Nonetheless, several issues could be distilled for which Linked Data would not necessarily provide the solution. These are as follows:

Unawareness of the available data
Researchers are unaware of what data is available to them. This hinders them during the early stages of their research, as they fail to explore data outside the scope of their current search.

Uncertainty about how to locate relevant data
Researchers deem it difficult to get a proper understanding of the actions required to find and access the data they are looking for. Instead, they run into issues of ambiguity with the terms they use to search.

Inability to effectively distil relevant data

Researchers feel overwhelmed when faced with an exhaustive list of *potentially relevant* data. To determine what data is relevant and what is not, they need to thoroughly examine each entry in the list.

Incompleteness of data

Researchers are confronted with incomplete data, which manifests itself as gaps in the information they access. As they cannot ascertain whether the missing data is of importance, this hinders them from trusting the data set as a whole.

8.2 Data Understanding

A vital step in developing a data mining solution is to properly understand the data upon which it will be applied. In the case of ARIADNE, this data would consist of Linked Archaeological Data. Unfortunately, little of this data has become available as of yet. Therefore, Linked Data from several different archaeological repositories was inspected instead. These data were chosen for their almost disjoint characteristics, thus hopefully providing good representations of the different facets that ARIADNE might have to incorporate. Assuming this is the case, several observations can be made:

Different ontologies

Data sets from different sources differ in the ontologies they use. This occurs less within data sets from the same source. In addition, different versions of the same ontology can be observed between data sets. Furthermore, some of the ontologies used were specifically created for the repository where the data is hosted.

Structural variation

Data sets from the same source appeared to contain relatively little variation; most of a data set's concept descriptions follow approximately the same structural schema. That is, nearly all descriptions in a data set use roughly the same set of properties, which are specified using the same ontologies. The inverse appears true between the data sets from different sources, which all have their own distinct schemas.

Structurally-flat graphs

It seems quite common for data sets to have a graph representation, which is relatively flat in structure. This appears to originate from a scarceness of URIs per resource, either due to a resource only having a small number of properties, or due to an extensive use of literals. In the latter case, numerous occurrences of unnecessary use have been observed. That is, literals were used to denote property values for which URIs are available.

Strong Dependency on descriptive values

All studied data sets appear to depend strongly on descriptions. In most cases, these descriptions provide crucial information and thus cannot be omitted without a significant loss of knowledge. However, such descriptions are often represented as a single literal, thus being little more than unstructured text.

Concurrent RDF statements

Data sets were observed in which both the correct and incorrect statements for the same concept were kept available. That is, instead of updating falsified statements, an erratum was supplied. Due to this construction, duplicate and possibly conflicting entries might surface.

8.3 Recommendations

Generally, the developer of a typical Data Mining solution is supplied with a generous amount of data from which the exploration might reveal potentially relevant patterns. After careful inspection of the data currently available, relevant patterns can likely be generalized to the entirety of the data. Unfortunately, the minimal amount of data current available through ARIADNE prevents such sequence of events to take place. Therefore, it would be rather unlikely to successfully generalize any discovered pattern to the large amount of data that, one day, will be accessible through ARIADNE. Instead, a more-generic approach is suggested, such that its workings are ensured regardless of the exact characteristics of the future data.

Based on the study of both the domain and data generated by the domain, as well as on practical constraints with respect to time and resources, two data mining solutions were chosen which were deem the most-feasible and suitable for implementation within the ARIADNE framework. These are:

Hypothesis Generation (7.1)

The official project proposal of ARIADNE mentions the ability to detect patterns in archaeological data or related data and applications within the ARIADNE infrastructure. Data-mining methods are capable of detecting such patterns. Interesting and potentially relevant subsets of these patterns can then be presented to users as starting points for forming new research hypotheses. This may already be known to the researcher, or it might be something they are interested in exploring further. The interestingness of patterns will be determined algorithmically on the basis of predefined criteria and user feedback. To facilitate this, a user interface should provide access to a data mining backend. Initially, this might best be integrated within a text-based query interface such as SPARQL or similar. At a later stage, a wrapper for the graphical user interface should be made. This will allow multiple hypotheses to be presented in non-intrusive ways, thereby allowing users to quickly scan in potentially valuable directions.

Data Quality Analysis (7.5)

Two aspects that reflect poorly on the quality of data are the occurrence of gaps and errors in the knowledge contained therein. In case of the former, filling these voids involves predicting the most-likely resource, link, or literal. In the latter case, these errors typically include anomalies within the data which cannot be explained by any of the discovered patterns alone. Depending on the likelihood of them being erroneous, the detected errors could be suggested for removal or tagged as dubious. Alternatively, they could be replaced by a prediction of the correct value. This could reuse the earlier-mentioned prediction method and data mining backend.

8.4 Roadmap

The sequel to this report, i.e. Deliverable 16.3, will present the final results of the applicability and feasibility of data mining within the ARIADNE framework. To this end, the aforementioned recommendations will be explored and experimented with further. This process will consist of several phases.

Following this report, a more extensive study into the recommended topics will first be performed. This will assist us in narrowing down the list of possible options to only those that we believe possess the most potential. The remaining options will subsequently be implemented into our experimental environment on site where they will thoroughly be tested on various linked archaeological data. The result of these tests will determine whether the selected options are suitable for integration within ARIADNE. Based on the experience gained during the current study, we expect that most of the possible options will need to be adapted to suit both the data and the user's needs. If, for some reason, none of these options are found to be suitable, a custom solution will be developed instead.

Once the selected options have been successfully implemented within the experimental environment, internal evaluation rounds will be organized during which domain enthusiasts and experts of varying levels of expertise will be asked to experiment with the implementation. Here, we expect the groups to be comprised of students, junior and senior archaeological researchers, and local data and repository managers. This will additionally provide the input needed for the development of (elements of) a graphical user interface. Similarly as before, an iterative scheme will be followed.

The final phase will consist of potentially implementing the data mining solutions into the ARIADNE infrastructure. This would be followed by extensively experimenting on the various data accessible through ARIADNE. This implementation will be improved upon further during a series of iterative and open evaluation sessions for the remainder of the ARIADNE project.

Bibliography

Aloia, N, C Meghini, D Gavrilis, and C Papatheodorou. *ARIADNE Catalogue Data Model.* Deliverable, ARIADNE, 2014.

Amin, A, J Van Ossenbruggen, L Hardman, and A van Nispen. "Understanding cultural heritage experts' information seeking needs." *The 8th ACM/IEEE-CS joint conference on Digital libraries.* ACM, 2008. 39-47.

Anyanwu, K, A Maduko, and A Sheth. "SemRank: ranking complex relationship search results on the semantic web." *14th international conference on World Wide Web .* ACM, 2005. 117-127.

Artz, D, and Y Gil. "A survey of trust in computer science and the semantic web." *Web Semantics: Science, Services and Agents on the World Wide Web*, 2007: 58-71.

Balmin, A, V Hristidis, and Y Papakonstantin. "Objectrank: authority-based keyword search in databases." *VLDB*, 2004: 564-575.

Baxter, M. *Statistics in archaeology.* London: Arnold, 2003.

Berendt, B, A Hotho, and G Stumme. "Towards semantic web mining." *The Semantic Web—ISWC.* Springer Berlin Heidelberg, 2002. 264-278.

Berendt, B., Hotho, A, D Mladenic, M Van Someren, M Spiliopoulou, and G Stumme. "A roadmap for web mining." *Web to semantic web*, 2004: 1-22.

Bi, S, S Xue, Y Xu, Pei, and A. "Spatial Data Mining in Settlement Archaeological Databases Based on Vector Features." *Fuzzy Systems and Knowledge Discovery.* Jinan Shandong: IEEE, 2008. 277-281.

Bicer, V, T Tran, and A Gossen. "Relational kernel machines for learning from graph-structured RDF data." *The Semantic Web: Research and Applications*, 2011: 47-62.

Bizer, C, and R Oldakowski. "Using Context- and Content-Based Trust Policies on the Semantic Web." *13th International World Wide Web Conference.* New York, NY: ACM Press, 2004. 228-229.

Bizer, C, T Heath, and T Berners-Lee. "Linked data-the story so far." *International journal on semantic web and information system* 3, no. 5 (2009).

Bloehdorn, S, and Y Sure. "Kernel methods for mining instance data in ontologies." 2007: 58-71.

Borgwardt, K M, N N Schraudolph, and S Vishwanathan. "Fast computation of graph kernels." *Advances in neural information processing systems*, 2006: 1449-1456.

Bray, T, J Paoli, C M Sperberg-McQueen, E Maler, and F Yergeau. "Extensible Markup Language (XML) 1." *W3C.* November 26, 2008. www.w3.org/TR/xml.

Brickley, D. "Basic Geo Vocabulary." *W3C.* February 1, 2006. www.w3.org/2003/01/geo.

Briscoe, T. "Lexical issues in natural language processing." *Natural language and speech*, 1991: 39-68.

Buneman, P. "Semistructured data." *The sixteenth ACM SIGACT-SIGMOD-SIGART Principles of database systems.* ACM, 1997. 117-121.

Campinas, S, T E Perry, D Ceccarelli, R Delbru, and G Tummarello. "Introducing rdf graph summary with application to assisted sparql formulation." *23rd International Workshop on Database and Expert Systems Applications.* 2012.

Castells, P, M Fernandez, and D Vallet. "An adaptation of the vector-space model for ontology-based information retrieval." *Knowledge and Data Engineering.* IEEE Transactions, 2007. 261-272.

Charno, M, S Jeffrey, C Binding, D Tudhope, and K May. "From the Slope of Enlightenment to the Plateau of Productivity: Developing Linked Data at the ADS." *40th Annual Conference of Computer Applications and Quantitative Methods in Archaeology.* Southampton: Amsterdam University Press, 2012. 216-223.

Chen, M S, J Han, and P S Yu. "Data mining: an overview from a database perspective." *Knowledge and data Engineering*, 1996: 866-883.

Cleave, J P. *A study of logics.* Oxford University Press, 1991.

Codd, E F. "A relational model of data for large shared data banks. ." *Communications of the ACM*, 1970: 377-387.

Codd, E F, S B Codd, and C T Salley. *Providing OLAP (on-line analytical processing) to user-analysts: An IT mandate.* Codd and Date, 1993.

Cohen, W W, and Z Kou. "Stacked graphical learning: approximating learning in markov random fields using very short inhomogeneous markov chains." Technical report, 2006.

Conolly, J, and M Lake. *Geographical Information Systems in Archaeology.* Cambridge: Cambridge University Press, 2006.

Cripps, P, et al. *CRMarchaeo: the Excavation Model.* CIDOC CRM, 2014.

Cyganiak, R, and D Reynolds. "The RDF Data Cube Vocabulary." *W3C.* January 2014, 2014. www.w3.org/TR/vocab-data-cube.

d'Amato, C, N Fanizzi, and F Esposito. "Inductive learning for the Semantic Web: What does it buy?" *Semantic Web*, 2010: 53-59.

De Kleijn, M, N van Manen, J Kolen, and H Scholten. "Towards a User-centric SDI Framework for Historical and Heritage European Landscape Research." *International Journal of Spatial Data Infrastructures Research*, 2014: 1-35.

Di Ludovico, A, and G Pieri. "Artificial Neural Networks and ancient artefacts: Justifications for a multiform integrated approach using PST and Auto-CM models." *Archeologia e calcolatori*, 2011: 91-128.

Dimitropoulos, H, et al. "AITION: a scalable platform for interactive data mining." *Scientific and Statistical Database Management*, 2012: 646-651.

Doerr, M, and K Schaller. "The Dream of a Global Knowledge Network - A new Approach." *ACM Journal on Computers and Cultural Heritage*, 2008.

Doerr, M, K Schaller, and M Theodoridou. "Itegration of complementary archaeological sources." *Computer Applications and Quantitative Methods in Archaeology.* Prato, Itally , 2004.

Dubin, D. "The most influential paper Gerard Salton never wrote." *Library Trends*, 2004: 748-764.

Earl, G, T. T Sly, and D D Wheatley. "Archaeology in the Digital Era." *Computer Applications and Quantitative Methods in Archaeology.* Southampton: Amsterdam University Press, 2014.

Etcheverry, L, and A A Vaisman. "Enhancing OLAP analysis with web cubes." *Semantic Web: Research and Applications*, 2012: 469-483.

Etcheverry, L, and A A Vaisman. *QB4OLAP: A New Vocabulary for OLAP Cubes on the Semantic Web.* R1210LAC004, 2012.

Etter, D, and C Domeniconi. "SemRank: Semantic Rank Learning for Multimedia Retrieval." 2014.

Fanizzi, N, C d'Amato, and F Esposito. "Conceptual clustering and its application to concept drift and novelty detection." Munich: Springer Berlin Heidelberg, 2008.

Fayyad, U M. "Data mining and knowledge discovery: Making sense out of data." *IEEE Intelligent Systems* 11, no. 5 (1996): 20-25.

Fayyad, Usama, Gregory Piatetsky-shapiro, and Padhraic Smyth. "From Data Mining to Knowledge Discovery in Databases." *AI Magazine* 17 (1996): 37-54.

Fentress, E. *Register of Online Archaeological Databases.* Deliverable, ARIADNE, 2014.

Fisher, D H. "Knowledge acquisition via incremental conceptual clustering." *Machine learning* 2, no. 2 (1987).

Franz, T, A Schultz, S Sizov, and S Staab. "Triplerank: Ranking semantic web data by tensor decomposition." 2009: 213-228.

Freitas, A, E Curry, J G Oliveira, and S O'Riain. "Querying heterogeneous datasets on the linked data web: Challenges, approaches, and trends." *Internet Computing, IEEE*, 2012: 24-33.

Friedman, J H. "Data Mining and Statistics: What's the connection?" *Computing Science and Statistics* 29, no. 1 (1998): 3-9.

Garshol, L M. " Metadata? Thesauri? Taxonomies? Topic maps! Making sense of it all." *Journal of information science* 4, no. 30 (2004): 378-391.

Gärtner, T. "A survey of kernels for structured data." *ACM SIGKDD Explorations Newsletter*, 2003, 5 ed.: 49-58.

Getoor, L, and B Taskar. *Introduction to statistical relational learning.* MIT press, 2007.

Golbeck, J, B Parsia, and J Hendler. "Trust networks on the semantic web." *Journal of Web Semantics*, 2003: 238-249.

Gombos, G, and A Kiss. "SPARQL query writing with recommendations based on datasets." *Information and Knowledge Design and Evaluation*, 2014: 310-319.

Gruber, E, G Bransbourg, S Heath, and A Meadows. "Linking Roman Coins: Current Work at the American Numismatic Society." *40th Annual Conference of Computer Applications and Quantitative Methods in Archaeology.* Southampton: Amsterdam University Press, 2012. 249-258.

Gupta, M, Y Sun, and J Han. "Trust analysis with clustering." *20th international conference companion on World Wide Web .* ACM, 2011. 53-54.

Hagood, J. "A brief introduction to data mining projects in the humanities." *Bulletin of the American Society for Information Science and Technology* 38, no. 4 (2012): 20-23.

Hastie, T, R Tibshirani, J Friedman, T Hastie, J Friedman, and R Tibshirani. *The elements of statistical learning.* New York: Springer, 2009.

He, X, and M Baker. "xhRank: Ranking Entities for Semantic Web Searching." *Fifth International Conference on Advances in Semantic Processing.* 2011. 62-68.

Heath, T, and Christian Bizer. *Linked Data: Evolving the Web into a Global Data Space.* Morgan & Claypool Publishers, 2011.

Hogan, A, A Harth, and S Decker. "Reconrank: A scalable ranking method for semantic web data with context." 2006.

Hollander, Hella, and Maarten Hoogerwerf. *D13.1: Service Design.* ARIADNE, 2014.

Huang, B, A Kimmig, L Getoor, and J Golbeck. "Probabilistic soft logic for trust analysis in social networks." *International Workshop on Statistical Relational AI.* 2012. 1-8.

Huang, Y, and V Tresp. "Accessing the Semantic Web via Statistical Machine Learning." *ESWC 2012 Tutorial.* May 2012, 2012. http://www.dbs.ifi.lmu.de/~huang/eswc2012tutorial/ESWC2012-TutorialV10.pdf.

Huang, Y, and V Tresp. *Relation Prediction in Semantic Domains using Multivariate Prediction.* Munich, Germany: Siemens AG, 2010.

Huang, Y, V Tresp, H Kriegel, and P. "Multivariate prediction for learning in relational graphs." *Workshop: Analyzing Networks and Learning With Graphs.* 2009.

Huang, Y, V Tresp, M Bundschus, A Rettinger, and H P Kriegel. "Multivariate prediction for learning on the semantic web." *Inductive Logic Programming*, 2011: 92-104.

Huber, R, H Ramoser, K Mayer, H Penz, and M Rubik. "Classification of coins using an eigenspace approach." *Pattern Recognition Letters*, 2005: 61-75.

Huggett, J. "Disciplinary Issues: Challenging the Research and Practice of Computer Applications in Archaeology." *40th Annual Conference of Computer Applications and Quantitative Methods in Archaeology.* Southampton: Amsterdam University Press, 2012. 14-24.

Isaksen, L, G Earl, K Martinez, S Keay, and N Gibbins. "Linking archaeological data." *International Conference on Computer Applications and Quantitative Methods in Archaeology.* 2009.

Isaksen, L, K Martinez, N Gibbins, G Earl, and S Keay. "Linking archaeological data." *CAA*, 2009.

Isaksen, L, K Martinez, N Gibbins, Graeme Earl, and S Keay. "Interoperate with whom? Fomality, Archaeology and the Semantic Web." *WebScience.* Raleigh, NC, 2010.

Kantardzic, M. *Data mining: concepts, models, methods, and algorithms.* John Wiley & Sons, 2011.

Karasik, A, I Sharon, U Smilansky, and A Gilboa. "Typology and classification of ceramics based on curvature analysis." *Computer Applications and Quantitative Methods in Archaeology.* 2004. 472-475.

Kellar, M, C Watters, and K M Inkpen. "An exploration of web-based monitoring: implications for design." *The SIGCHI conference on Human factors in computing systems.* ACM, 2007. 377-386.

Khan, M A, G A Grimnes, and A Dengel. "Two pre-process operators for improved learning from semanticweb data." *First Rapidminer Community Meeting and Conference.* 2010.

Kiefer, C, A Bernstein, and A Locher. "Adding data mining support to SPARQL via statistical relational learning methods." *The Semantic Web: Research and Applications*, 2008: 478-492.

Kintigh, K. "Quantitative methods designed for archaeological problems." In *Quantitative Research in Archaeology: Progress and Prospects*, by M S Aldenderfer, 135-150. Newbury Park, NJ: Sage, 1987.

Knobbe, A J. *Multi-relational data mining.* Ios Press, 2006.

Kobylinski, L, and K Walczak. "Data mining approach to classification of archaeological aerial photographs." *Intelligent Information Processing and Web Mining*, 2006: 479-487.

Kolda, T G, and B W Bader. "Tensor decompositions and applications." *SIAM review* 51, no. 3 (2009): 455-500.

Koller, D, and N Friedman. *Probabilistic graphical models: principles and techniques.* MIT press, 2009.

Kramer, K, R Q Dividino, and G Gröner. "SPACE: SPARQL Index for Efficient Autocompletion." *International Semantic Web Conference.* 2013. 157-160.

Krogel, M A, S Rawles, F Železný, P A Flach, N Lavrač, and S Wrobel. "Comparative evaluation of approaches to propositionalization." *13th International Conference on ILP.* Szeged: Springer Berlin Heidelberg, 2003. 197-214.

Kurgan, L A, and P Musilek. "A survey of Knowledge Discovery and Data Mining process models." *The Knowledge Engineering Review*, 2006: 1-24.

Lavrač, N, A Vavpetič, L Soldatova, I Trajkovski, and P K Novak. "Using ontologies in semantic data mining with segs and g-segs." *Discovery Science*, 2011: 165-178.

Lavrac, N, and S Dzeroski. *Relational Data Mining.* Springer, 2001.

Linderholm, J, and P Geladi. "Classification of archaeological soil and sediment samples using near infrared techniques." *NIR news*, 2012: 6.

Locher, A. *SPARQL-ML: Knowledge Discovery for the Semantic Web.* Thesis University of Zurich, University of Zurich, 2007.

Maali, F, J Erickson, and P Archer. "Data Catalogue Vocabulary (DCAT)." *W3C.* January 16, 2014. www.w3.org/TR/vocab-dcat.

Maedche, A, and V Zacharias. "Clustering ontology-based metadata in the semantic web. In." *Principles of Data Mining and Knowledge Discovery.* Springer Berlin Heidelberg, 2002. 348-360.

Maimon, O Z, and L Rokach. *Data mining and knowledge discovery handbook.* New York: Springer, 2005.

Mani, Inderjeet, and Mark T Maybury. *Advances in Automatic Text Summarization.* Cambridge: MIT Press, 1999.

Martinez, K, and L Isaksen. "The Semantic Web Approach to Increasing Access to Cultural Heritage." *Revisualizing Visual Culture.* Varnham, 2010. 29-44.

May, K. *Hypermedia Research Unit - CIDOC CRM-EH Ontology.* n.d. http://hypermedia.research.southwales.ac.uk/kos/CRM/ (accessed November 02, 2014).

Mendes, P N, M Jakob, A García-Silva, and C Bizer. "DBpedia spotlight: shedding light on the web of documents." *7th International Conference on Semantic Systems.* ACM, 2011. 1-8.

Metaxas, O, H Dimitropoulos, Y Ioannidis, and M Paedigree. "AITION: A scalable KDD platform for Big Data Healthcare." *Biomedical and Health Informatics.* IEEE, 2014. 601-604.

Middleton, S E, N R Shadbolt, and D C De Roure. "Ontological user profiling in recommender systems." *ACM Transactions on Information Systems (TOIS) 22.1*, 2004: 54-88.

Narasimha, V, P Kappara, R Ichise, and O P Vyas. "LiDDM: A Data Mining System for Linked Data." *CEUR Workshop Proceedings: Linked Data on the Web.* 2011.

Nickel, M, V Tresp, and H Kriegel. "A three-way model for collective learning on multi-relational data." *28th international conference on machine learning.* 2011. 809-816.

Nolle, M, H Penz, M Rubik, K Mayer, I Hollander, and R Granec. "Dagobert-a new coin recognition and sorting system." *International Conference on Digital Image Computing, Techniques, and Applications.* Sydney: CSIRO Publishing, 2003. 329-338.

Novak, P K, A Vavpetic, I Trajkovski, and N Lavrac. "Towards semantic data mining with g-segs." *11th International Multiconference Information Society.* 2009.

OGC GeoSPARQL - A Geographic Query Language for RDF Data. Specification, Open Geospatial Consortium, 2012.

Padawitz, P. *Computing in Horn clause theories.* Springer Publishing Company, 2012.

Parsaye, K. "Surveying Decision Support: New Realms of Analysis." *Database Programming and Design*, 1996: 26-33.

Paulheim, H, and J Fümkranz. "Unsupervised generation of data mining features from linked open data." *International conference on web intelligence, mining and semantics.* ACM, 2012. 31.

Potoniec, J, and A Lawrynowicz. "RMonto: ontological extension to RapidMiner." *ISWC.* 2011. 1-4.

—. "RMonto-towards KDD workflows for ontology-based data mining." *eCML PKDD.* 2011b. 11.

Prud'hommeaux, E, and A Seaborne. "SPARQL Query Language for RDF." *W3C.* January 18, 2008. www.w3.org/TR/rdf-sparql-query.

Pu, L, and B Faltings. "Understanding and improving relational matrix factorization in recommender systems." *7th ACM conference on Recommender systems.* ACM, 2013. 41-48.

ragimov, D, K Hose, T B Pedersen, and E Zimányi. "Towards Exploratory OLAP over Linked Open Data–A Case Study." 2014: 18.

Ramezani, R, M Saraee, and M A Nematbakhsh. "Finding association rules in linked data, a centralization approach." *Iranian Conference on Electrical Engineering.* IEEE, 2013. 1-6.

Rapidminer SemWeb. n.d. https://code.google.com/p/rapidminer-semweb (accessed September 12, 2014).

Rettinger, A, U Lösch, V Tresp, C d'Amato, and N Fanizzi. "Mining the semantic web." *Data Mining and Knowledge Discovery*, 2012: 613-662.

Richards, J D. "Archaeology, e-publication and the semantic web." *ANTIQUITY-OXFORD* 80, no. 310 (2006): 970-979.

Ristoski, P, and H Paulheim. "A Comparison of Propositionalization Strategies for Creating Features from Linked Open Data." *Linked Data for Knowledge Discovery*, 2014: 6-17.

Ristoski, P, C Bizer, and H Paulheim. "Mining the web of linked data with rapidminer." *International Semantic Web Conference.* 2014.

Rocha, C, D Schwabe, and M P Aragao. "A hybrid approach for searching in the semantic web." *13th Iinternational conference on World Wide Web.* ACM, 2004. 374-383.

Salton, G, A Wong, and C S Yang. "A vector space model for automatic indexing." *Communications of the ACM*, 1975: 613-620.

Schölkopf, B, and A J Smola. *Learning with kernels: Support vector machines, regularization, optimization, and beyond.* MIT press, 2002.

Selhofer, Hannes, and Guntram Geser. *D2.1: First Report on Users' Needs.* Salzburg: ARIADNE, 2014.

Sen, P, G Namata, M Bilgic, L Getoor, B Galligher, and T Eliassi-Rad. "Collective classification in network data." *AI magazine* 29, no. 3 (2008).

Shadbolt, N, W Hall, and T Berners-Lee. "The semantic web revisited." *Intelligent Systems* (IEEE) 3, no. 21 (2006).

Signore, O. "Representing knowledge in archaeology: from cataloguing cards to semantic web." *Archeologia e calcolatori*, no. 20 (2009): 111-128.

Singla, P, and P Domingos. "Entity resolution with markov logic." *Sixth International Conference on Data Mining.* IEEE, 2006. 572-582.

Sloman, S A, and D A Lagnado. *The Problem of Induction.* Cambridge University Press, 2005.

Stumme, G, A Hotho, and B Berendt. "Semantic Web Mining and the Representation, Analysis, and Evolution of Web Space." *RAWS.* Prague, 2005. 1-16.

Stumme, G, A Hotho, and B Berendt. "Semantic web mining: State of the art and future directions." *Web semantics: Science, services and agents on the world wide web* 2, no. 4 (2006): 124-143.

Szomszor, M, et al. "Folksonomies, the semantic web, and movie recommendation." 2007.

Taranto, C, N Di Mauro, and F Esposito. "Learning in probabilistic graphs exploiting language-constrained patterns." In *New Frontiers in Mining Complex Patterns*, 155-169. Springer Berlin Heidelberg, 2013.

The CIDOC Conceptual Reference Model. n.d. www.cidoc-crm.org (accessed November 02, 2014).

Tous, R, and J Delgado. "A vector space model for semantic similarity calculation and OWL ontology alignment." *Database and Expert Systems Applications*, 2006: 307-316.

Tresp, V, M Bundschus, A Rettinger, and Y Huang. "Towards machine learning on the semantic web." In *Uncertainty Reasoning for the Semantic Web I*, by P G da Costa, et al., 282-314. Springer Berlin Heidelberg, 2008.

Tudhope, D, K May, C Binding, and A Vlachidis. "Connecting archaeological data and grey literature via semantic cross search." *Internet Archaeology* 30 (2011).

van der Maaten, L, P Boon, G Lange, H Paijmans, and E Postma. "Computer vision and machine learning for archaeology." *Computer Applications and Quantitative Methods in Archaeology.* 2006. 112-130.

van Harmelen, Frank, Grigoris Anoniou, Paul Groth, and Rinke Hoekstra. *A Semantic Web Primer.* Cambridge: MIT Press, 2012.

Vishwanathan, S V N, N N Schraudolph, R Kondor, and K M Borgwardt. "Graph kernels." *Journal of Machine Learning Research*, 2010: 1201-1242.

Wagtendonk, A J, P Verhagen, S Soetens, K Jeneson, and M De Kleijn. "Past in Place: The Role of Geo-ICTin Present-day Archaeology." In *Geospatial technology and the role of location in science*, by H J Scholten and N van de Velde, R van Manen, 96. London: Springer, 2009.

Wells, J J, et al. "Web-based discovery and integration of archaeological historic properties inventory data: The Digital Index of North American Archaeology." *Literary and Linguistic Computing* 3, no. 29 (2014): 349-360.

Whallon, R. "Simple statistics." In *In Quantitative Research in Archaeology: Progress and Prospects*, by M S Aldenderfer, 135-150. Newbury Park, NJ: Sage, 1987.

Witten, Ian H, Eibe Frank, and Mark A Hall. *Data Mining: Practical Machine Learning Tools and Techniques.* 3rd. Burlington: Morgan Kaufmann, 2011.

Appendix A Reasoning with Logic

Two commonly-applied forms of logical reasoning are deductive and inductive inference. The Semantic Web inherently facilitates the former, whereas DM and ML inherently facilitate the latter. Both forms differ in their capabilities, which makes their fusion an interesting possibility. In other words, by incorporating DM into the SW, inductive inferences can be made in conjunction with deductive inferences. To clarify this statement, we will discuss each of these aspects in slightly more depth next.

A.1 Reasoning by Deduction

On the SW, the ability to carry out deductive reasoning is being provided by Ontology languages. These languages are more than simply data-description languages. In fact, they are knowledge-representation languages; formal (logic) systems that allow the inference of implicit knowledge from explicit knowledge (van Harmelen, et al. 2012, Stumme, Hotho and Berendt 2006). For instance, given that we know the Coliseum to be in Rome, and given that we know Rome is in Italy, we may infer the additional knowledge that the Coliseum is in Italy.

The formalizations used by LD are grounded in **Horn Logic**; a subset of predicate logic (van Harmelen, et al. 2012). Horn Logic provides a rule-based system with which one may infer a consequence based on an antecedent (Cleave 1991, Padawitz 2012). Within the context of LD, this consequent is an RDF triple, whereas the antecedent consists of one or more of such triples. For instance, let G be an RDF graph with n triples, and let $A \in G$ and C be the triples in that graph and a newly-formed triple respectively. The inherent ability of LD to reason deductively may then be described as in Equation B-1. That is, given known triples A_1 through A_n, triple C may be inferred. This sequence of terms is called a **definite clause**.

$$A_1, \dots, A_n \rightarrow C \qquad \textit{Equation A-1}$$

As a subset of predicate logic, Horn Logic inherits two unique properties (van Harmelen, et al. 2012, Padawitz 2012, Cleave 1991); **soundness** and **completeness**. The former guarantees that no false statements will be inferred, provided that 1) all terms in the antecedent hold and 2) that the whole definite clause is valid. Completeness guarantees that every term that holds can be derived through reasoning. Together, these two properties provide proof of the validity of the statements in the system (van Harmelen, et al. 2012). Within the context of LD, this ensures that, assuming that all triples in the data store are valid, all inferred knowledge is valid as well.

A.2 Reasoning by Induction

Deductive systems allow one to derive new statements from existing statements, as well as prove the validity of these existing statements (Cleave 1991, van Harmelen, et al. 2012). In other words, the truth value of a piece of knowledge may be derived through reasoning about existing knowledge. When the quality of that existing knowledge lacks however, e.g. when it is incomplete, it may limit the system's ability to derive these truth values (d'Amato, Fanizzi and Esposito 2010, Rettinger, et al. 2012). A solution to this might be provided by inductive reasoning (Huggett 2012, Rettinger, et al. 2012).

Inductive reasoning can be regarded as a method of **approximation** (Rettinger, et al. 2012, Sloman and Lagnado 2005). Hence, it is unable to prove the validity of statements. Instead, it may provide a probability of a statement being true based on the **evidence** is has at its disposal. For instance, let G be an RDF graph with n triples, and let $A \in G$ and C be the triples in that graph and a new triple respectively (Equation B-2). The probability of triple C being true, given the triples A_1 through A_n, may now be inferred.

$$P(C|A_1, \dots, A_n) \qquad \textit{Equation A-2}$$

The methods used in the field of DM are based on inductive reasoning (Rettinger, et al. 2012). Therefore, any representation of the data's knowledge these methods generate is merely an approximation. This approximation however, provides an inherit ability to generalize over and even beyond the original data (Witten, Frank and Hall 2011, Kantardzic 2011, Chen, Han and Yu 1996).

A.3 Logic Reasoning within the Semantic Web

The strength of the Semantic Web originates from the formal logics on which it is built. While powerful, relying solely on these formalisms may have its drawbacks (Rettinger, et al. 2012, Tresp, et al. 2008). The deductive-reasoning capabilities that it brings forth are completely based upon axiomatic prior knowledge, and thus are unable to exploit any regularities in the data which have not been formulated as ontological knowledge. In addition, its ability to reason with uncertainty has long been a hurdle, which has only recently gotten attention from researchers. Even when turning a blind eye towards these yet-unsolved aspects, it would prove quite a challenge to find a real-world data set that is able to fulfil the formal requirements. Such sets tend to be quite large in size, however, thus scaling up the amount of data to reason with; yet another hurdle LD has yet to be overcome.

Semantic-Web Mining (SWM) is an umbrella term, which denotes the area of DM that focusses on its applicability to LD. As such, it offers the prospect of **inductive reasoning within the Semantic Web**. Incorporating such an approach may (partially) remedy the drawbacks of pure deductive reasoning. That is, an inductive approach is, in the optimal case, much more capable of reasoning with (partially) incomplete and uncertain data, as well as processing large volumes of that data. Placed within the Semantic Web Stack (Figure A-1), it belongs on the same level as deductive reasoning.

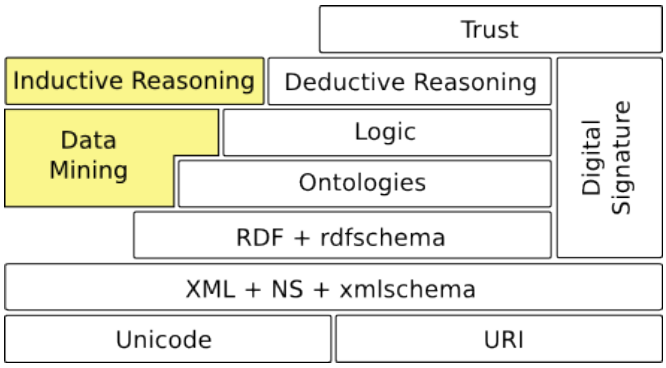

Figure A-1: Data Mining and Inductive Reasoning within the Semantic Web Stack.

Appendix B Vector Space Models

Vector Space Models (VSM) are a group of simple mathematical models generally known for their role in the field of Information Retrieval (IR) (Dubin 2004, Salton, Wong and Yang 1975). These models represent vectors in which every term corresponds to a distinct word in a document. That is, each VSM is a vector $\overrightarrow{D_i}$ with J terms, whereby every term d_{ij} represents a distinct word $j \in [0,\ J)$ of a document i (Equation C-1). Here, J denotes the number of unique words in document i.

$$\overrightarrow{D_i} = \langle\, d_{i0},\ d_{i1},\ d_{ij} \,\rangle \qquad \textit{Equation B-1}$$

$$\overrightarrow{D_i} = \langle\, w_0 d_{i0},\ w_1 d_{i1},\ w_j d_{ij} \,\rangle \qquad \textit{Equation B-2}$$

In its most basic form, each term is a binary value which denotes whether the corresponding document contains at least one occurrence of the word that this term represents. However, it is more common to multiply each term with a weight that represents the importance of that term or word (Equation C-2). For instance, a popular weight metric is the **Term Frequency-Inverse Document Frequency**, which considers how often a term occurs in a document and how this relates to its 'natural' occurrence.

By computing the similarity between two vectors, the similarity between the two documents they belong to is computed as well. As this is operation requires few resources, two or more documents can be compared both swiftly and cheaply (Figure B-1). Similarly, a set of documents can efficiently be searched for those that are most-similar to a certain query. More specifically, by translating both documents and query to vectors, their similarity can efficiently be computed. Furthermore, this similarity may form the basis by which an **order of relevance** can be determined.

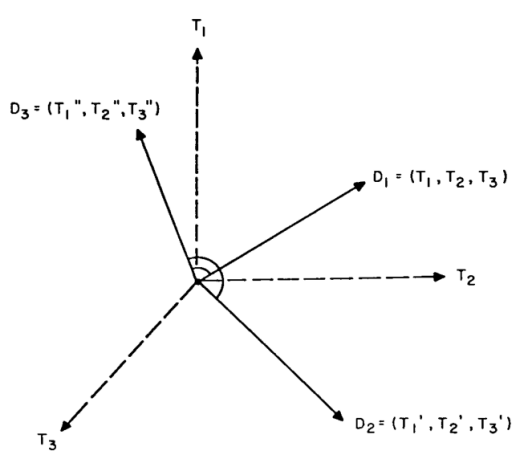

Figure B-1: Three vectors consisting of three terms depicted in their corresponding Vector Space (Salton, Wong and Yang 1975).

Appendix C Learning Methods for Semantic Web Mining

Semantic-Web Mining offers a wide range of potential methods, most of which are rather experimental. Of these methods (Knobbe 2006, Rettinger, et al. 2012, Berendt, et al. 2004), two are featured prominently in recent literature, namely Propositionalization and Statistical Relational Learning. Of these, the core concepts will be discussed next. In addition, we will briefly look at the promising and rather new Kernel Methods, which have recently become quite popular within ML communities.

C.1 Propositional Learning

Recall that propositional data are assumed to be independent and identically distributed, hence allowing for statistical machine learning algorithms to be applied. Given that these assumptions do not necessarily hold for LD, applying them anyway would likely result in false conclusions. Instead of abandoning these proven methods however, an alternative would be to temporarily convert LD to propositional data; a process known as **propositionalization** (Ristoski and Paulheim 2014, Tresp, et al. 2008).

A well-known method during propositionalization is **factorization**, which involves the decomposition of an object into a product of smaller objects; its factors. When put together again, these factors return (an estimation of) the original object. Within the context of LD, it entails the translation of graph data to propositional data, and vice versa. While in this proposition format statistical ML can be applied.

C.1.1 Relational Matrices

A matrix is a two-dimensional array arranged in n rows and m columns. Each point (i, j) with $i \in n$ and $j \in m$ is an element of the matrix. When two or more matrices are needed to describe a single data set, these are often denoted as relational matrices. Note that this strongly resembles the relational data model as is used in relational database.

Translating LD into relational matrices (Figure C-1) is a fairly straightforward procedure (Tresp, et al. 2008, Pu and Faltings 2013, Rettinger, et al. 2012). Each matrix represents a single predicate in the data set. Hence, the number of required matrices equals the number of predicates. Next, of each triple containing a certain predicate, its *subject* and *object* are placed on row i and column j of the corresponding matrix, respectively. Within that matrix, the element (i, j) now contains a value of 1.0, thereby representing that the predicate holds with respect to the corresponding *subject* and *object.*

After propositionalization, each (i, j) of a matrix contains either a value of 1.0 (holds) or 0.0 (otherwise) (Pu and Faltings 2013, Rettinger, et al. 2012). Factorization is then applied to determine the latent features that are hidden between the entities. This is similar to the well-known statistical technique of Principal Component Analysis (PCA). Once split up, the matrices are multiplied again, thereby creating an estimation of the original matrix. However, where before some of the entities had the value 0.0, they now have a value between 0.0 and 1.0, thereby representing confidence values that the corresponding statement holds.

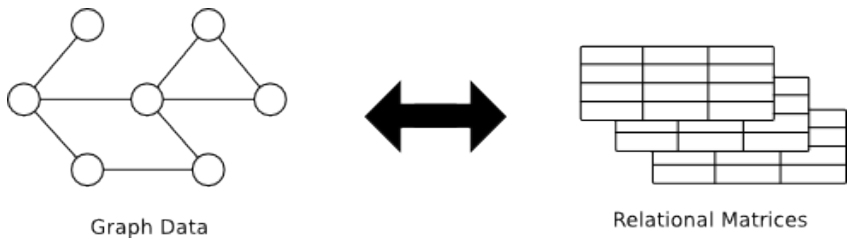

Figure C-1: Propositionalization of graph data to relational matrices. Note that, for clarity, only three distinct RDF predicates have been depicted as relational matrices, whereas the graph depicts a maximum of nine.

An advantage of using relational matrices is that the processes of applying propositionalization and factorization are fairly straightforward. While many different approaches to factorizing a matrix exist, a technical detail omitted here, it constitutes a proven method overall that is in use at many companies (Pu and Faltings 2013, Rettinger, et al. 2012). Furthermore, once an RDF graph has gone through this process, the reconstructed statements that were previously unknown could be integrated as (weighted) triples (Tresp, et al. 2008). Care should be taken however, as perhaps not every unknown statement is justified. Another aspect of which one should be wary is that all statements of a certain predicate are estimated in a single step. Extending this over the entire graph provides a performance that scales proportionally to the number of predicates, as well as to their frequency. That is, every additional triple requires an additional entry in the corresponding (possibly not yet existing) matrix.

C.1.2 Tensors

Tensors may be thought of as the generalization of the matrix, thereby allowing for an arbitrary number of modes, called *orders*. Differently put; a matrix can be regarded as a second-order tensor. Depending on the number of orders N, a tensor consists of points $(i, j, \dots, n)$ with $i, j, \dots, n \in N$. Note however, that orders are independent of the spatial dimensions.

During propositionalization (Figure C-2), each graph is translated to a single tensor (Kolda and Bader 2009, Rettinger, et al. 2012, Nickel, Tresp and Kriegel 2011). Given that each triple consists of three entities, those tensors will be of the third order. Of these orders, the first will contain all subjects, the second all predicates, and the third all objects of the graph. Assuming indices i, j, k for the first, second,

and third order respectively, each element (i, j, k) will now contain the value 1.0 if the statement holds, and 0.0 otherwise.

Factorization is then applied to determine the latent features that are hidden between the entities. Once split up, the tensors are multiplied again, thereby creating an estimation of the original tensor (Kolda and Bader 2009, Rettinger, et al. 2012). However, where before some of the entities had the value 0.0, they now have a value between 0.0 and 1.0, thereby representing confidence values that the corresponding statement holds.

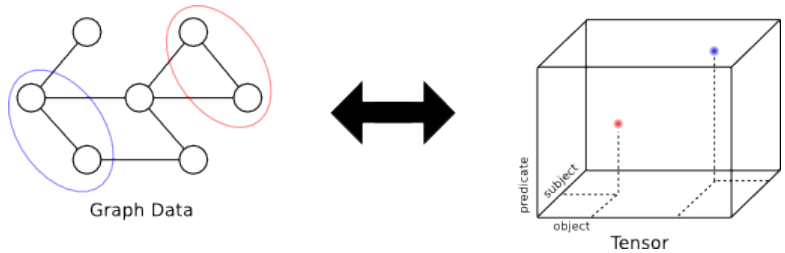

Figure C-2: Propositionalization of graph data to a third-order tensor in a three-dimensional vector space. Note that, for clarity, only two RDF statements have been depicted in the tensor, whereas the graph depicts nine.

An advantage of using tensors is the simplicity of applying propositionalization and factorization. In addition, tensors permit the inclusion of contextual information (Rettinger, et al. 2012), as well as allowing for collective learning such that it is less dependent on the explicit aggregation of information. Furthermore, once an RDF graph has gone through this process, the reconstructed statements that were previously unknown could be integrated as (weighted) triples (Tresp, et al. 2008). Care should be taken however, as perhaps not every unknown statement is justified. Another aspect of which one should be wary is that all statements of the whole graph are estimated in a single step. Nevertheless, tensors tend to scale rather well, despite their large size. That is, each order ranges proportionally to the number of corresponding entities within the graph.

C.2 Statistical Relational Learning

Statistical Relational Learning is an umbrella term, which encompasses a large number of various methods to represent, reason, and learn in domains with complex relational and rich **probabilistic structures** (Getoor and Taskar 2007, Rettinger, et al. 2012). Typically, these methods are based on either logic- or frame-based formalisms, such as rules or graphical models respectively.

C.2.1 Inductive Logic Programming

Inductive Logic Programming (ILP) encompasses methods that attempt to learn logical (Horn) clauses directly from relational data (Lavrac and Dzeroski 2001, Getoor and Taskar 2007, Tresp, et al. 2008, Maimon and Rokach 2005). These clauses typically consist of conjunctions of positive and negative

logical atoms, which together, can be seen as Logic Programs. In addition, these Logic Programs allow for the integration of valid background (domain) knowledge as well.

Given a set of positive and negative atoms of target relation p, as well as given a set of background relations q_i, the task is to learn a definition of relation p that is consistent and complete (Lavrac and Dzeroski 2001, Getoor and Taskar 2007, Maimon and Rokach 2005). That is, this definition should be able to explain all specified positive and negative atoms. Within the context of LD, both types of atoms can be thought of as triples that either do or do not hold. In that case, the outcome of a single clause would then constitute the truth value of a hypothesized triple.

As an example, consider wanting to determine whether a colour difference in a layer of excavated soil are the remains of an ancient water well. Given numerous examples of colour differences that were found to be such a water well, and given numerous more examples of those that were not, a new rule could be learned that discriminates between the two cases. A very crude version of such a rule, or Logic Program, might resemble Equation D-1, which evaluates the colour difference CD as being a well ($V = t$) if its radius is larger than 50 units and its depth larger than 200 units, as well as having a difference D in colour with respect to the surrounding area SA that exceeds a value of 1.42 units.

$$waterWell(CD, V = t) \leftarrow \qquad \textit{Equation C-1}$$

$$radius(CD) > 50 \wedge depth(CD) > 200$$

$$\wedge\, colourDifference(CD, SA, D) \wedge D > 1.42$$

A big advantage of ILP is the experience it has on (multi-) relational DM (Getoor and Taskar 2007, Lavrac and Dzeroski 2001, Maimon and Rokach 2005), which, while not the same, shares many of the hurdles that are also present with graph data. Another advantage is its logical representation, which provides a strong expressive power and which fits naturally to the formal logics behind LD. This fit allows for an easy integration of background knowledge, as well as allowing for an equally easy integration of inferred knowledge into the original graph. However, while some variants do exist, this new knowledge is mainly limited to statements that either do or do not hold. Another disadvantage is the need for multiple positive and negative atoms per relation, which in the case of LD, is challenging due to the often-existing sparseness in RDF graphs. Moreover, the need for negative atoms translated into a need for explicitly specified triples that state that a relation does not hold; a practice that is rare in a system that adheres to the Open-World Assumption.

C.2.1.1 Propositional ILP

In addition to relational learning, ILP also has a strong relation with propositional learning. In fact, it was in the area of ILP where the term *propositionalization* was originally used. Within that context, it constitutes the translation of first-order clauses into features to which statistical ML algorithms can be applied. However, this process may result in the loss of information, and thus is said to exchange accuracy for efficiency (Krogel, et al. 2003).

With propositionalization (Figure C-3), a set of Boolean features is sought whereby each feature can be defined as a corresponding clause (Krogel, et al. 2003, Lavrac and Dzeroski 2001, Tresp, et al. 2008). Given n features, a propositionalization of a relational-learning problem is a set of n clauses, with each clause constituting one or more logical literals. These literals are derived from the relational background knowledge. Once all features have been defined, they are evaluated with respect to an instance, thus resulting in a sequence of n Boolean values.

Within the field of ILP, there are two approaches to propositionalization (Krogel, et al. 2003, Lavrac and Dzeroski 2001); either completely or partially. With complete propositionalization, all knowledge contained within a data set is translated to feature definitions, whereas with the partial variant this is done only for the most-relevant subset of this knowledge. Therefore, in case of the latter, certain knowledge is lost. Moreover, determining which features are of importance is typically accomplished through heuristics, thus introducing assumptions.

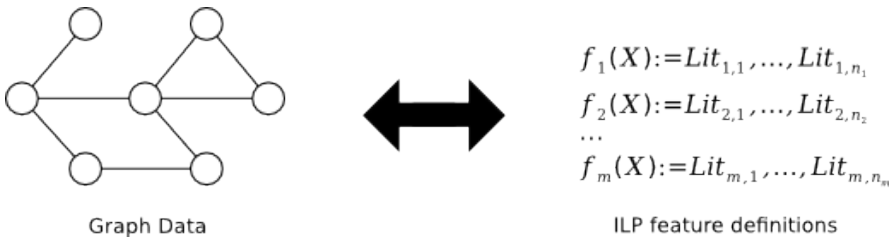

Figure C-3: Propositionalization of graph data to a set of ILP feature definitions.

An advantage of propositional ILP are that, in most cases, it is a very robust and effective method (Krogel, et al. 2003, Lavrac and Dzeroski 2001, Tresp, et al. 2008). Only a few data sets are known to exist for which this does not hold. Another benefit is that the partial variant may easily be tailored to one's needs by specifying the number of features. This latter feature can additionally be regarded as a disadvantage, as it may quickly lead to lost knowledge. Moreover, it might introduce assumptions that may bias any future reasoning.

C.2.2 Relational Graphical Models

Probabilistic Graphical Models (PGM) are a general framework (Figure C-4) to represent complex real-world phenomena over a high-dimensional space by the combination of probability theory and logical structures (Getoor and Taskar 2007, Cohen and Kou 2006, Koller and Friedman 2009). Therefore, PGMs are able to reason with uncertainty, as well as with dependencies between entities. While numerous kinds exist, the majority of the PGMs can easily be depicted by either a directed or undirected graph, such as a Bayesian and Markov model, respectively. Within this graph, the nodes map to domain variables and the edges correspond to direct probabilistic interactions between these variables. Furthermore, PGMs have a fixed graphical structure, which limits their ability to reason about a varying number of entities in a variety of configurations.

Relational Graphical Models (RGM) are a specific kind of PGMs (Figure C-4) that extend the PGM framework with a flexible graphical structure, as well as with concepts of objects, their properties, and relations between them (Getoor and Taskar 2007, Tresp, et al. 2008, Lavrac and Dzeroski 2001). This separation is similar to that of propositional and relational logics. Irrespective, whether the relations hold is encoded by their corresponding binary variables, which are represented as nodes in the RGM. Within the context of LD, these variables denote the potential RDF triples and *not* the nodes of the corresponding RDF graph, which are either resources or literals. More specifically, a variable with the value 1.0 would denote that the corresponding triple holds, whereas the value 0.0 would denote the opposite. Any value in between would indicate a certainty value.

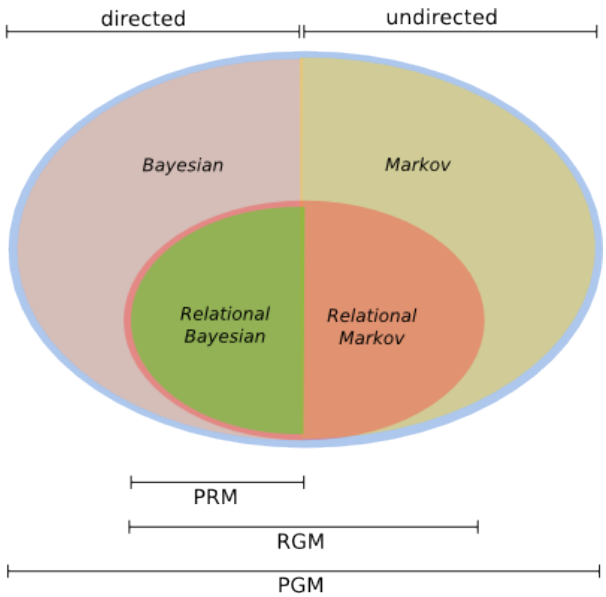

Figure C-4: Venn diagram of the hierarchy within the PGM framework. Note that the cursive terms within the boundaries denote that class' most-common model.

As with PGMs, RGMs may be either directed or undirected (Figure C-4). In case of the former, these are generally known as Probabilistic Relational Models (PRM) them (Getoor and Taskar 2007, Tresp, et al. 2008, Lavrac and Dzeroski 2001, Rettinger, et al. 2012). In addition, the type of relation and its direction are assumed to be known; an assumption intended to simplify the model. An extension to this additionally considers two types of structural uncertainty:

Reference uncertainty concerns the case in which a relation, and only one of its two members, is known. That is, it is unknown whether a certain entity is dependent upon one or more other entities, and if so, which those other entities are.

Existence uncertainty concerns the case in which the relation between two or more entities is unknown. That is, given two entities, it is unknown whether one depends on the other.

As an example, consider translating a simple RDF graph (Figure C-5 Left) to a PRM (Figure C-5 Right). Here, the former involves an anomaly discovered within the soil, which was found to be an ancient water well. This conclusion was based upon the radius and depth of the anomaly, thus making these measurements dependencies for that conclusion. Therefore, a PRM would represent the corresponding triples, i.e. *Soil Anomaly has radius 60* and *Soil Anomaly has depth 250* as parent nodes of the concluding triple *Soil Anomaly a Water Well*.

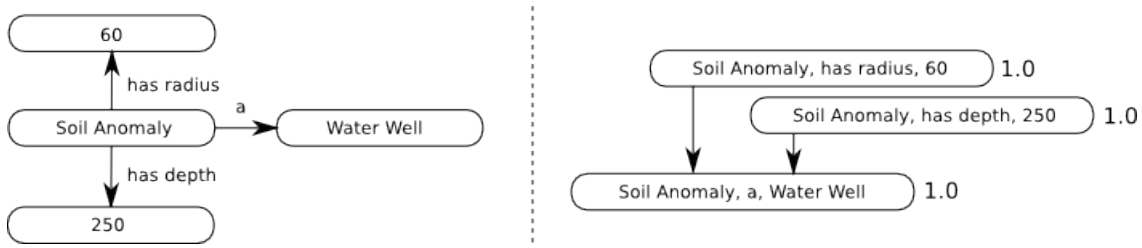

Figure C-5: Left) a simple RDF graph concerning an anomaly in the soil that once constituted a water well. Right) a PRM of the RDF graph with every triple being true.

An advantage of RGMs is the flexible but powerful expressiveness of their graphical representation (Getoor and Taskar 2007, Tresp, et al. 2008, Lavrac and Dzeroski 2001, Rettinger, et al. 2012). Being grounded in a sound statistical framework, they allow for direct learning from graph data without the need for propositionalization. In addition, ontology background knowledge can be integrated, and inferred knowledge can be incorporated into the RDF graph as weighted triples. This makes it particularly well suited to exploratory data analysis. Furthermore, it possesses the ability to learn and perform inference in large networks. However, both tasks tend to be expensive, as computational requirements scale with the number of statements whose truth value is known. Alternative approaches have been suggested, which limit the learning and reasoning to relevant subgraphs only. These approaches however, are still fairly new and experimental. Another challenge still largely unresolved is how to deal with missing data. Finally, one should note that RGMs are mainly limited to learning and predicting truth values of RDF statements.

C.3 Kernel Methods

Kernel methods are a group of techniques that try to solve mathematically complex ML problems by translating them into more-mathematically-workable ML problems (Schölkopf and Smola 2002). The function that performs this translation is called the *kernel*. How such a kernel is defined depends greatly on the characteristics of the problem at hand. In fact, many different kernels exist, each designed with a specific goal in mind.

Mathematically speaking[45]; given a problem specified in *input space* $\mathcal{X}$, a mapping Φ is defined which translates points in $\mathcal{X}$ to *feature space* $\mathcal{H}$ (Equation D-2). A kernel k is then defined such that it maps input vectors x and x' from $\mathcal{X}$ to $\mathcal{H}$ using mapping Φ (Equation D-3).

$$\Phi : \mathcal{X} \rightarrow \mathcal{H} \qquad \textit{Equation C-2}$$

$$k(x, x') \longmapsto \langle \Phi(x), \Phi(x') \rangle \qquad \textit{Equation C-3}$$

Amongst the most popular methods that use kernels are Support-Vector Machines (SVM) (Schölkopf and Smola 2002, Gärtner 2003). SVMs are commonly used to perform classification tasks (Figure C-6). However, they can additionally be used for either clustering or regression. By incorporating kernels, which they refer to as applying the ***kernel trick***, SVMs are able to mitigate several of the issues that are encountered by traditional ML algorithms. One prime example of such an issue is local minima, which cause an optimization process to stall. Moreover, the kernel trick allows SVMs to learn non-linear separation boundaries.

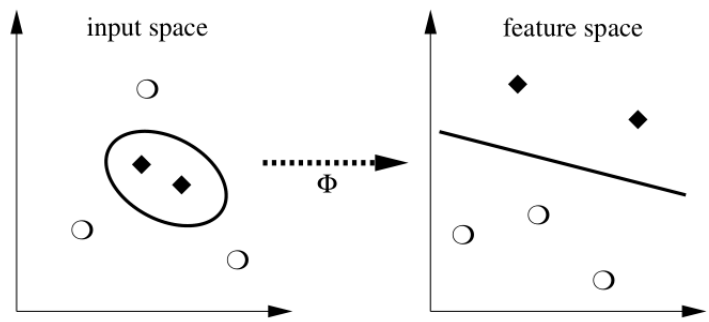

Figure C-6: Example of how SVMs translate complex problems (left), here a binary-classification problem, to more-workable problems (right) (Schölkopf and Smola 2002).

[45] For simplicity, the terminology chosen is the one from commonly-used in SVMs. Regardless, the principles remain the same.

An advantage of using kernel methods is they do not restrict their arguments to solely vector-type data. Instead, they may be defined on almost any type of data (Bloehdorn and Sure 2007, Gärtner 2003). As a result, they can be applied directly to heterogeneous and interconnected data without the need for converting these data to vectors. A related advantage of kernels is their ability to simplify complex learning problems by mapping them to less-complex learning problems. Moreover, SVM, the field's most well-known method, is one of the most successful recent developments in ML. In fact, it has been shown to effectively solve learning problems which were left unsolved by ILP; a field with many more years of experience. However, most of the success stories on kernel methods stem from academic research. So far, very little experience has been achieved outside of those confines. Due to the field's current popularity however, this disadvantage may eventually be nullified.

C.3.1 Kernels for Structured Data

There are two common approaches when developing kernels for structured data. These approaches concern the decision to use either model-driven or syntax-driven kernels (Gärtner 2003). Here, the former is often applied when either background knowledge or states[46] are of importance. In contrast, the latter emphasizes the semantics within a data set. These kernels typically involve rules, trees, and graphs.

Kernels developed for learning on the SW often fall into the group of **Graph Kernels**. Each of these graph kernels constitutes a function that translates a graph to an element of which the similarity with other elements can easily be computed. Unfortunately, this translation step is generally rather resource expensive, especially when isomorphism[47] has to be taken into account (Gärtner 2003, Borgwardt, Schraudolph and Vishwanathan 2006, Vishwanathan, et al. 2010). A compromise is to use approximate kernels instead. One well-known and simple example of these is a Random-Walk Graph kernel, which computes the translation of a graph by randomly walking over its vertices. Two graphs G and G' can subsequently be checked on similarity by comparing their corresponding random walks.

Another kind of kernel that is appropriate to learn from the SW is the **Clause Kernel** (Bicer, Tran and Gossen 2011). Each clause kernel contains an ILP clause which corresponds to a feature of the corresponding RDF graph. This approach adds the advantage of dynamically defining kernels based on the likelihood of them being able to explain a set of provided examples. Furthermore, several of these kernels may subsequently be combined into one composite kernel, thereby improving efficiency. In addition, combining kernels provides resilience to sparse data.

[46] A state involves a snapshot of a data set at a specific time. Applying an operation, e.g. add or remove, to a state results in a new state.
[47] Isomorph graphs only differ in the enumeration of their vertices.

Appendix D Sample of Archaeological Scenarios

Several junior and senior researchers, as well as a couple of Ph.D. candidates, were invited to participate in a brainstorming session at the Faculty of Archaeology of Leiden University as well as at the VU University Amsterdam. Those participating were asked to write down an archaeological research scenario, which they deemed difficult with the current information infrastructure. In addition, participants were instructed to assume that all required data was readily available within this infrastructure.

The following scenarios were submitted (translated from Dutch):

1) *The need to retrieve all literature about individual finds from very (often obscure) local journals about a particular excavation (i.e. the late bronze age settlement in Bovenkarspel).*
2) *The need for information on plants which are useful for humans, both as food, oil, medicinal or as material (for creating ropes, buckets etc.) found in Holocene contexts dated to Late prehistory (i.e. Late Neolithic, Bronze Age and Iron Age).*
3) *All publications from excavated settlements which are dated to the Bronze Age and located on a specific geomorphological unit, in an area from Denmark to North France.*
4) *All images (images and drawings) with metadata about the dating and archaeological context from prehistoric traps made of willow. In addition a list of persons who researched these artefact types.*
5) *Overview of all Neolithic axes made of stone, and in particular flint found in a Roman context. (i.e. found during excavations, not individual finds).*
6) *The reliability and usefulness of C14 dating depends on a variety of factors (i.e. stratigraphical position, soil disturbances, type of sample etc.). This information is noted on separate sample forms. It would be very useful if this information were accessible on to the level of individual C14 identifiers. (e.g. www.lumid.nl provides TL information, but not as LOD).*
7) *Functionality to retrieve all the GIS data from a specific archaeological area (e.g. all GIS data from Roman excavations on the Kops Plateau in Nijmegen).*
8) *All published radiometric dated (in years) material (with id numbers) on Neanderthal sites in France.*
9) *All types of arrowheads dated to the middle Neolithic period.*
10) *All characteristics of hand axes from the South East of the Netherlands.*
11) *All information on the origins of the Levallois technique. (site locations X,Y coordinates).*
12) *Information on the influence of the pH value soils on the conservation conditions of charcoal.*
13) *All archaeological contexts in which broken flint axes dated from the Middle Neolithic period are found. (i.e. settlements, funerary, individual finds etc.).*
14) *All sources, both ethnographic, historic and archaeological, in which land clearances by fire is mentioned with regard to their influence the landscape vegetation.*
15) *Functionality to automatically search for keyword synonyms (for example in Google scholar).*

16) *Need to find publication from a specific period in which specific contexts and find conditions are present (i.e. burial sites with bones intact, dated to the Neolithic period).*
17) *Functionality to automatically search for indicators of fire usage in Paleolithic Europe. (e.g. charred, burnt, heated, burned, craquelure, etc.).*
18) *Functionality to narrow the search results down to archaeologically relevant results; domain specific (e.g. querying "Bakers' Hole" or "Belvédère" in Google gives a lot of archaeological irrelevant results.*
19) *Images from middle Paleolithic arrowheads with traces of reuse from several (internationally distributed) datasets, giving problems with language and semantics etc.).*
20) *From a specific type of arrowheads the length and width visualized in a scatter diagram. How much does the users own measurements dataset deviate from datasets with comparable arrowheads? (visualisation of both datasets combined).*
21) *Comparing 13C/15N isotope analysis from bones, which have to be subdivided into types of animals and type of bone (teeth, skull, spines etc.) related to the area, of a particular site to other sites.*
22) *A lot of excavation material from already excavated sites is stored in regional depots. To find specific artefacts one has to go through all boxes (sometimes up to 300). This is very time consuming, it would therefore be very useful if you can automatically query analogue lists.*
23) *All animals that currently live on open arctic steppe areas and which provide a minimum of 50 kilos of meat. All Paleolithic sites which have bones from these types of animals as well as the contexts in which they have been found.*
24) *Combining archaeological data from several different sources without having to worry about the differences between these data. Currently, this poses a large problem as sharing data and combining it with data from external sources is rarely considered when constructing a database.*
25) *Evaluating the quality of the data before considering using it. Is it biased in some way and, if so, to what degree? What assumptions are made? Are there any outliers? Are there uncertainties and/or contradictions? What is the statistical significance of the outcome?*
26) *Display a Harris Matrix of the data. Instead of creating one manually, automatically generate one based on the provided data.*