UNIVERSIDADE DE LISBOA

FACULDADE DE CIÊNCIAS

DEPARTAMENTO DE INFORMÁTICA

Breaking rules: taking Complex Ontology Alignment beyond rule-based approaches

Beatriz Fonseca de Lima

Mestrado em Ciência de Dados

Dissertação orientada por:

Prof.ª Doutora Cátia Luísa Santana Calisto Pesquita

Prof. Doutor Daniel Pedro de Jesus Faria

2021

Acknowledgements

I want to thank my supervisors, professors Cátia Pesquita and Daniel Faria, for their reliable guidance and close attention to this study. They are authentic role models for their work methodology and dedication to Science, the scientific community, and their students. I feel very fortunate to have learned and worked with such a brilliant duo, who are not only experts in their field but also very good at passing down their knowledge.

I am thankful to my family and friends who have supported me throughout my academic experience. A special thank to Diogo, who has consistently met me with kind words and cheers. To God, for He is the source of all my life and strength.

I want to express my gratitude to my Data Science peers, who have always been available to provide a helping hand and share their knowledge and skills.

I extend my thanks to Lu Zhou, for kindly providing the source code used in the OAEI evaluation.

Finally I want to thank the support of Fundação para a Ciência e a Tecnologia, which provided the funding under LASIGE Research Unit, ref. UIDB/00408/2020 and ref. UIDP/00408/2020, and by project SMILAX (ref. PTDC/EEI-ESS/4633/2014).

Abstract

As ontologies are developed in an uncoordinated manner, differences in scope and design compromise interoperability. Ontology matching is critical to address this semantic heterogeneity problem, as it finds correspondences that enable integrating data across the Semantic Web. One of the biggest challenges in this field is that ontology schemas often differ conceptually, and therefore reconciling many real-world ontology pairs (e.g., in geography or biomedicine) involves establishing complex mappings that contain multiple entities from each ontology. Yet, for the most part, ontology matching algorithms are restricted to finding simple equivalence mappings between ontology entities.

This work presents novel algorithms for Complex Ontology Alignment based on Association Rule Mining over a set of shared instances between two ontologies. Its strategy relies on a targeted search for known complex patterns in instance and schema data, reducing the search space. This allows the application of semantic-based filtering algorithms tailored to each kind of pattern, to select and refine the most relevant mappings.

The algorithms were evaluated in OAEI Complex track datasets under two automated approaches: OAEI's entity-based approach and a novel element-overlap–based approach which was developed in the context of this work. The algorithms were able to find mappings spanning eight distinct complex patterns, as well as combinations of patterns through disjunction and conjunction. They were able to efficiently reduce the search space and showed competitive performance results comparing to the State of the Art of complex alignment systems.

As for the comparative analysis of evaluation methodologies, the proposed element-overlap–based evaluation strategy was shown to be more accurate and interpretable than the reference-based automatic alternative, although none of the existing strategies fully address the challenges discussed in the literature.

For future work, it would be interesting to extend the algorithms to cover more complex patterns and combine them with lexical approaches.

Keywords: Ontology Alignment, Ontology Matching, Complex Ontology Alignment, Association Rule Mining

Resumo

Os dados existem na Web em formatos muito heterogéneos. Os conceitos de "dados interligados" e "Web Semântica" emergiram num esforço para potencializar o uso dos dados em aplicações inteligentes. Sob este paradigma os dados encontram-se caracterizados, contextualizados e associados a outros dados, tornando-os acessíveis a operações humanas e automáticas. Neste contexto, os dados heterogéneos poderiam ser facilmente integrados e ser alvo de mineração por agentes inteligentes, sustentando diversas aplicações corporativas e campos de investigação multidisciplinares tal como as ciências da vida e do ambiente.

A integração semântica dos dados requer uma camada estrutural, como é o caso das ontologias, que oferecem uma descrição expressiva dos conceitos e o contexto em que eles se inserem e podem ser utilizados. Contudo, diferentes ontologias surgem como consequência de existirem diversas formas de modelar o mesmo domínio, dependendo do objetivo e granularidade pretendida. Isto, combinado com o facto de que as ontologias são tipicamente desenvolvidas de forma independente nos vários domínios do conhecimento, deu origem ao problema de heterogeneidade semântica.

O alinhamento de ontologias surge como uma solução para este problema de interoperabilidade, uma vez que permite estabelecer correspondências entre conceitos de duas ou mais ontologias. Contudo, atualmente estes sistemas apenas abordam correspondências simples, i.e. aquelas que relacionam diretamente entidades individuais das ontologias. Contudo, as diferenças conceptuais entre as ontologias podem ser de tal magnitude que exigem um tipo de correspondência mais expressiva, de modo a garantir que todas as transformações de conceitos necessárias à interoperabilidade das ontologias são satisfeitas.

As *correspondências complexas* são correspondências mais expressivas que compreendem não só entidades individuais das ontologias, mas também expressões que conjugam ou modificam estas entidades através de operadores de restrição, conjunção, disjunção, entre outros. A um conjunto de correspondências complexas denomina-se um alinhamento complexo.

O Alinhamento Complexo de Ontologias é então visto como um caminho para a integração semântica de dados, e a sua relevância tem sido reconhecida pela comunidade científica. Em 2018, a Ontology Alignment Evaluation Initiative (OAEI) implementou uma modalidade para alinhamentos complexos, tendo fornecido dados de elevada qualidade, alinhamentos de referência e protocolos de avaliação. Contudo, persistem alguns desafios no que diz respeito às estratégias de avaliação, uma vez que estas não se adequam inteiramente à natureza dos alinhamentos complexos. Para além disso, o Estado da Arte em algoritmos de alinhamento de ontologias está, na sua maioria, limitado a encontrar equivalências simples entre entidades.

As estratégias para alinhamento complexo de ontologias que existem atualmente dividem-se em duas famílias: métodos baseados em semelhança lexical e métodos estruturais. Os métodos lexicais estão limitados a casos em que os nomes das entidades são semelhantes lexicamente, o que não é verdade para muitas ontologias do mundo real. Os métodos estruturais são tipicamente estatísticos ou recorrem a técnicas de mineração de dados para explorar padrões nos indíviduos partilhados entre duas ontologias populadas. A intuição para estes métodos é que se existem relações recorrentes entre indivíduos e as classes e propriedades que os descrevem nas duas ontologias, estes padrões frequentes ou correlações indicam a possibilidade de uma relação semântica relevante entre essas mesmas entidades.

Esta dissertação propõe uma série de algoritmos devotos ao alinhamento complexo de ontologias, que são baseados em regras de associação obtidas da mineração de dados de indivíduos partilhados entre duas ontologias. A estratégia consiste numa procura direcionada de padrões complexos conhecidos *a priori*, em dados provenientes dos esquemas e instâncias das ontologias. Isto permite por um lado reduzir o espaço de procura, e, por outro, refinar os resultados pela aplicação de algoritmos de filtração baseados na semântica de cada um dos padrões.

Este trabalho também propõe uma nova estratégia automática de avaliação de alinhamentos complexos, baseada num alinhamento de referência. Esta abordagem *element-overlap* reflete o esforço esperado para corrigir manualmente um alinhamento, atribuindo uma pontuação a cada correspondência que o compõe consoante a sua semelhança a uma correspondência comtemplada no alinhamento de referência. Esta semelhança é calculada através de um índice de Jaccard entre os dois conjuntos que contêm os elementos de cada correspondência (incluindo operadores).

Os algoritmos propostos foram implementados num sistema de alinhamento de ontologias conhecido, AMLC, que possui métodos para o carregamento de ontologias e estruturas de dados eficientes que comportam esses dados. Em seguida, foram avaliados segundo uma implementação do protocolo de avaliação utilizado na OAEI, mas também segundo a estratégia de avaliação proposta neste trabalho, *element-overlap*. Os dados utilizados e alinhamentos de referência são também provenientes da OAEI.

Os resultados mostram que é possível encontrar correspondências abrangendo oito padrões complexos distintos, assim como combinações desses padrões através de operadores de conjunção e disjunção. A performance dos algoritmos mostrou-se comparável ao Estado da Arte.

O facto de que os algoritmos serem desenhados especificamente para cada padrão permite que haja um controlo fino sobre o processo de alinhamento e sobre os algoritmos de refinamento, filtro e agregação. Esta estratégia provou-se benéfica uma vez que os resultados após filtração e agregagação aumentaram significativamente a precisão enquanto que o impacto no *recall* foi mínimo.

Contudo, a maior limitação dos algoritmos de alinhamento propostos reside também na sua natureza particionada aos padrões conhecidos, uma vez que isso implica que seja desenvolvido um algoritmo por cada padrão, colocando maior peso na implementação. É por essa razão também que os algoritmos deste trabalho se encontram limitados atualmente a padrões s:s e s:c, não apresentando resultados de performance favoráveis em *datasets* ricos em correspondências c:c.

Foram encontradas algumas diferenças nos resultados obtidos relativamente às duas estratégias de avaliação automática. Procedeu-se então a uma análise comparativa exaustiva que revelou que a abordagem *element-overlap* proposta é mais precisa e interpretável do que a alternativa, ainda que nenhuma

das estratégias atualmente existentes solucione por completo os desafios na avaliação discutidos na literatura. A estratégia de avaliação proposta possui várias limitações que advêm do facto de se sacrificar a precisão de avaliação pela escalabilidade, contudo, obter uma versão significativamente mais precisa, que, por exemplo, implementasse raciocínio dedutivo, não é uma tarefa trivial. Isto porque as correspondências complexas frequentemente incluem expressões que vão para além do reino de semântica DL, comprometendo a decidibilidade do problema.

Os algoritmos foram também avaliados manualmente, obtendo uma precisão global de 75%. A análise específica para cada padrão mostrou que os algoritmos não foram capazes de encontrar alguns dos padrões esperados, contudo, encontraram outros que não estavam contemplados na referência, com precisão elevada. Estes resultados sugerem que os alinhamentos de referência não contemplam todas as correspondências válidas entre duas ontologias e realçam a importância de estabelecer métricas de avaliação que sejam simultaneamente automáticas e considerem vários graus de correção.

A performance dos algoritmos de alinhamento foi também comparada à de uma implementação simples de um método de mineração de regras de associação tradicional, o FP-Growth. Esta implementação não foi capaz de gerar regras de associação para os conjuntos de dados de maior magnitude e complexidade, mostrando que a abordagem proposta nesta dissertação é capaz de reduzir de forma eficiente o espaço de procura e o tempo de corrida, produzindo resultados em tempo útil.

Em trabalhos futuros, seria interessante estender os algoritmos propostos neste trabalho a outros padrões complexos, tais como restrições de cardinalidade exatas, máximas e mínimas com valor superior a um, e padrões m:n, incluindo cadeias de propriedades. Para além disso, explorar outros conjuntos de dados, incluindo aqueles que não possuam dados de instâncias partilhadas entre as ontologias, com a ajuda de técnicas de alinhamentos de instâncias. Quanto às estratégias de avaliação, poderia ser proveitoso o uso de uma abordagem baseada em regras simples para comparar semanticamente as correspondências de forma a melhorar a precisão de avaliação.

Palavras Chave: Alinhamento de Ontologias, Alinhamento Complexo de Ontologias, Regras de Associação

Contents

List of Figures

List of Tables

Acronyms

AML AgreementMakerLight.

ARM Association Rule Mining.

AROA Association Rule-based Ontology Alignment.

CANARD Complex Alignment Need and A-box based Relation Discovery.

COA Complex Ontology Alignment.

CQA Competency Questions for Alignment.

EDOAL Expressive and Declarative Ontology Alignment Language.

KG Knowledge Graph.

ML Machine Learning.

OA Ontology Alignment.

OAEI Ontology Alignment Evaluation Initiative.

OWL Web Ontology Language.

PG-ARM Pattern-Guided Association Rule Mining.

RDF Resource Description Framework.

Chapter 1

Introduction

1.1 Context and Motivation

Data is ubiquitous on the Web and available in many heterogeneous formats. The concepts of Linked Data and Semantic Web emerged to potentiate the use of data in meaningful ways, by ensuring that it is properly described, contextualised, and linked to other data, so it is accessible to both humans and machines [4]. In this context, heterogeneous data could be smoothly integrated and mined by intelligent agents, fueling many enterprise applications and interdisciplinary research fields such as the life and environmental sciences [22].

The semantic integration of data requires a schema layer, such as ontologies, which provides a more expressive description of resources and the context in which they can be used. Different ontologies can model the same domain differently, depending on the intended application and the required granularity and expressiveness. This, combined with the widespread and typically uncoordinated development of ontologies in several domains, has led to the semantic heterogeneity problem [17].

The ontology alignment (or matching) field emerged to overcome this interoperability problem, by providing mappings interrelating the concepts between two or more related ontologies [17]. However, to date, most systems only cover simple ontology alignments that connect individual ontology entities directly through equivalence or subsumption relations [37]. This is often not sufficient to capture all data transformations required for the interoperability between ontologies, especially when they differ conceptually [36].

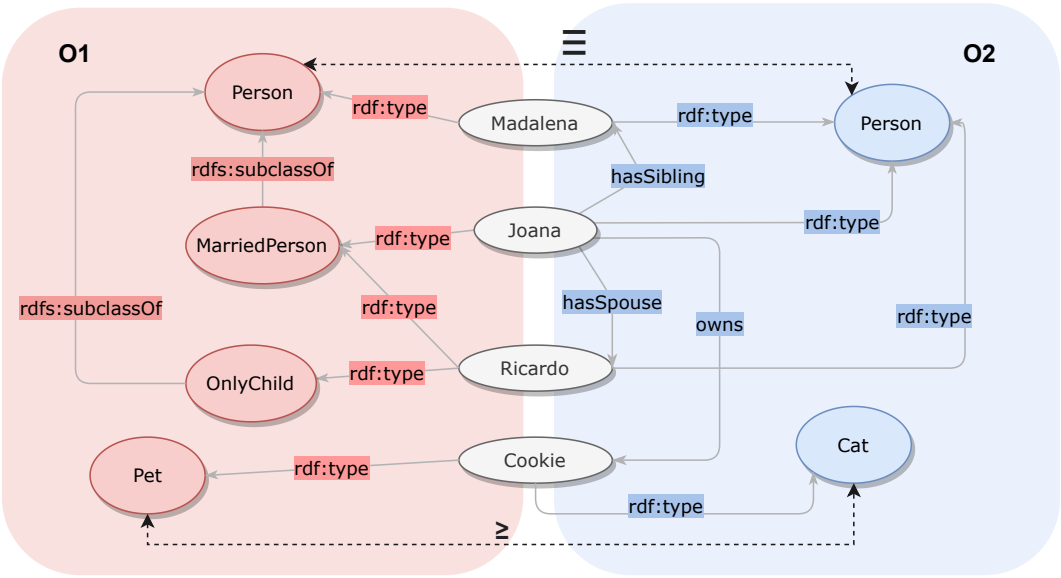


Figure 1.1: Ontology alignment given shared instance data. O1 and O2 represent two ontologies of intersecting domains. Coloured circles and rectangles represent ontology classes and data values, respectively. Full arrows are ontology properties and dashed arrows define the alignment relationship between ontology entities. Grey circles represent individuals shared between the two ontologies.

As an example, consider ontologies $O1$ and $O2$ (Figure 1.1) that model family relations under different perspectives. While $O1$ focuses on describing the main concepts as classes such as $O1 : MarriedPerson$, $O2$ does so using properties such as $O2 : hasSpouse$. In order to correctly transform data between the two ontologies, one would need to map the class $O1 : MarriedPerson$ to the class expression $O2 : hasSpouse$ some $Person$, i.e.:

$$O1 : MarriedPerson \equiv O2 : hasSpouse \text{ some } O2 : Person$$

Additionally, $O1$ and $O2$ differ in the level of detail used to describe the domain. In $O1$, the level of detail is such that it is possible to discriminate different types of people by the number of siblings they have. In $O2$, there is no such concept, only a more general one that describes that people can have siblings. A correct mapping in this case, would involve a cardinality restriction to that general concept.

$$O1 : OnlyChild \equiv O2 : hasSibling \text{ exactly } 1$$

The previous mappings are said to be *complex ontology mappings*, since at least one of the mapped entities is an expression rather than a simple ontology entity. They are more expressive than simple mappings (represented by dashed arrows in Figure 1.1), which fall short of describing the semantic meaning of the main concepts in this example.

Complex Ontology Alignment (COA) is then viewed as an essential tool for interoperability, semantic data integration and Data Science in general, as it aims to link data from different sources in a way that supports knowledge extraction. Doing complex alignment in an automated way is critical as ontologies often encompass thousands of concepts, making their manual alignment time-consuming and demanding of domain expertise.

The relevance of COA has been recognised by the ontology alignment community, which has been formulating complex alignment tasks and evaluation methodologies as part of the Ontology Alignment Evaluation Initiative (OAEI), since 2018. Nevertheless, there are still many challenges to overcome regarding benchmarks, evaluation, visualisation of alignments, among others [52]. Moreover, the performance of state-of-the-art alignment systems in complex tasks is not at a level where the alignments could be reliably used in real applications.

1.2 Objectives and Contributions

The goal of this work is to investigate data mining-based approaches for finding complex ontology mappings by searching for specific patterns defined *apriori*. The guiding hypothesis is that a targeted pattern-based approach that reduces the search space, will not only improve efficiency, but also performance by providing semantically sensible mappings.

According to [13], ontology alignment techniques can be divided into two categories: rule-based solutions and learning-based solutions. The first only exploit schema-level information in specific rules whereas the second may exploit instance information with machine-learning or statistical analysis. It is in that sense that this work proposes to go beyond rule based approaches, since, even though Association Rule Mining (ARM) is a rule-based machine learning technique, the strategy behind it relies on statistical measures and pattern mining over the ontologies schema and instance data.

This dissertation's main contributions include:

1. Development of PG-ARM, a suite of novel pattern-oriented and ARM-based complex alignment and filtering algorithms.
2. Comparative evaluation with state-of-the-art complex alignment systems using Ontology Alignment Evaluation Initiative (OAEI) datasets, under different evaluation modalities.
3. A novel element-overlap–based evaluation strategy for complex ontology alignments.
4. Fully automated implementation of the preprocessing step used in the entity-based evaluation strategy employed in the OAEI. Using this implementation, it was possible to perform the reference-based evaluation of OAEI's Populated Conference dataset, which is not presently contemplated in OAEI.
5. Poster presentations of preliminary results in the 6th LASIGE Workshop and the 15th International Workshop on Ontology Matching collocated in the 19th International Semantic Web Conference ISWC-2020.

6. Accepted poster in the 20th International Semantic Web Conference, ISWC-2021, titled “Pattern-Guided Association Rule Mining for Complex Ontology Alignment”.
7. Accepted full paper titled “Challenges of evaluating complex alignments” to the 16th International Workshop on Ontology Matching collocated in the 20th International Semantic Web Conference ISWC-2021.
8. Participation of the proposed alignment and filtering algorithms in the OAEI 2021’s Complex track.

1.3 Thesis Outline

This dissertation is organised as follows:

- The current chapter introduces the semantic heterogeneity problem and the complex ontology alignment field as a semantic data integration solution;
- Chapter 2 provides a formal explanation of key concepts required for understanding this work;
- Chapter 3 presents the State of the Art in the Complex Ontology Alignment field, regarding both matching and evaluation approaches; additionally, the use of ARM in other Semantic Web related tasks is discussed;
- Chapter 4 presents the algorithms for ontology alignment and ontology alignment evaluation developed in this work;
- Chapter 5 describes the data used in the evaluation of the proposed algorithms, presents their results under different evaluation modalities and discusses the challenges in evaluating complex alignments, comparing the proposed evaluation strategies to existing ones;
- Chapter 6 presents the main takeaways from this work, its limitations and the perspectives in terms of future work.

Chapter 2

Concepts

The data used in the scope of this work is not the most typical in Data Mining tasks, being enriched with semantic information. For this reason, the following sections delve into the particularities of this type of data and the Data Mining technique employed, Association Rule Mining.

2.1 Linked Data and Knowledge Graphs

Linked Data is a paradigm for publishing structured data on the Web interlinked to data from different sources [5]. It emerged as a re-interpretation of the Web of Documents, which consists of documents linked through hyperlinks [22]. As each document represents information in its own formats, machines are oblivious to the meaning and context of data, and automatic data processing approaches are not possible.

The Resource Description Framework (RDF) is the most popular format for Linked Data, which conveys that a specific data resource is connected to another resource through some kind of relationship, composing a triple *subject, predicate, object* [5].

The Semantic Web features Linked Data [4], enabling it to be shared and reused among applications. This translates into more powerful applications as their data sources are not restricted. Moreover, Data links can be used to integrate new data from different sources with ease, and automated knowledge discovery strategies can be employed [22].

In the last few years, *Knowledge Graph* (KG) has become a popular term to refer to a graph-based representation of large collections of data which are annotated to a semantic schema layer, such as ontologies [15]. In the case of RDF triples of Linked Data, the nodes in the graph represent RDF subjects and objects, and the edges represent RDF predicates.

2.2 Ontologies

Ontologies are key components of the Semantic Web, as they provide a semantic model of a domain so that it can be the object of automated reasoning. As defined in [17], ontologies "provide a vocabulary describing a domain of interest and a specification of the meaning of terms in that vocabulary". In other words, they specify the context and the semantic rules that apply to the vocabulary, allowing for the interpretation of those concepts through their logical axioms.

Using ontologies to represent domains reduces ambiguity and facilitates machine understanding [17]. They can serve as the semantic layer of smart information systems [42] or more recently, as the schema layer of Knowledge Graphs [15]. In the broader sense, their applications include named entity disambiguation, data interoperability and integration, enabling meaningful and intelligent queries over data on the Web and decision support.

Ontologies are often populated with *individuals*, which are instances at the data level, and include *literals*, such as strings or integers (see Figure 2.1). They are composed by *classes*, which represent categories or collections of individuals and *properties*, which establish a relation between an individual and a literal (*data property*) or another individual (*object property*). Ontologies intrinsically imply transitive closure, i.e. each given class is a subclass of not only its direct superclass but all the others that precede it.

The concepts of "Populated Ontology" and "Knowledge Graph" are often used interchangeably throughout the literature, being their separation somewhat ambiguous. Ehrlinger *et al.* [15] argue that the difference between the concepts resides in the fact that KG are typically large and provide additional features than ontologies, such as built-in strategies for knowledge derivation.

The use of ontologies does not impose a restricted or authoritarian way to model concepts, as users can create new ontologies at any given time, covering the same or intersecting domains, depending on the purpose and granularity required to their application. As ontologies are developed in an uncoordinated manner, some heterogeneity among them becomes inevitable. The heterogeneity may be in terms of syntax, when they are not expressed in the same ontology language, but also terminological, when different names are used to describe the same entities. Another source of heterogeneity, and usually the most difficult to overcome, is when ontologies differ conceptually: the same domain could be covered in different levels of detail and from different perspectives [17].

In this context, it can become difficult to define which concepts have the same meaning in the several existing ontologies. Furthermore, if different data sources use different ontologies to describe their data, one cannot ensure that the data is truly usable and integrated in a way that could allow intelligent queries, reasoning and the production of knowledge. In an effort to overcome this interoperability problem, ontology alignment strategies have been developed [17].

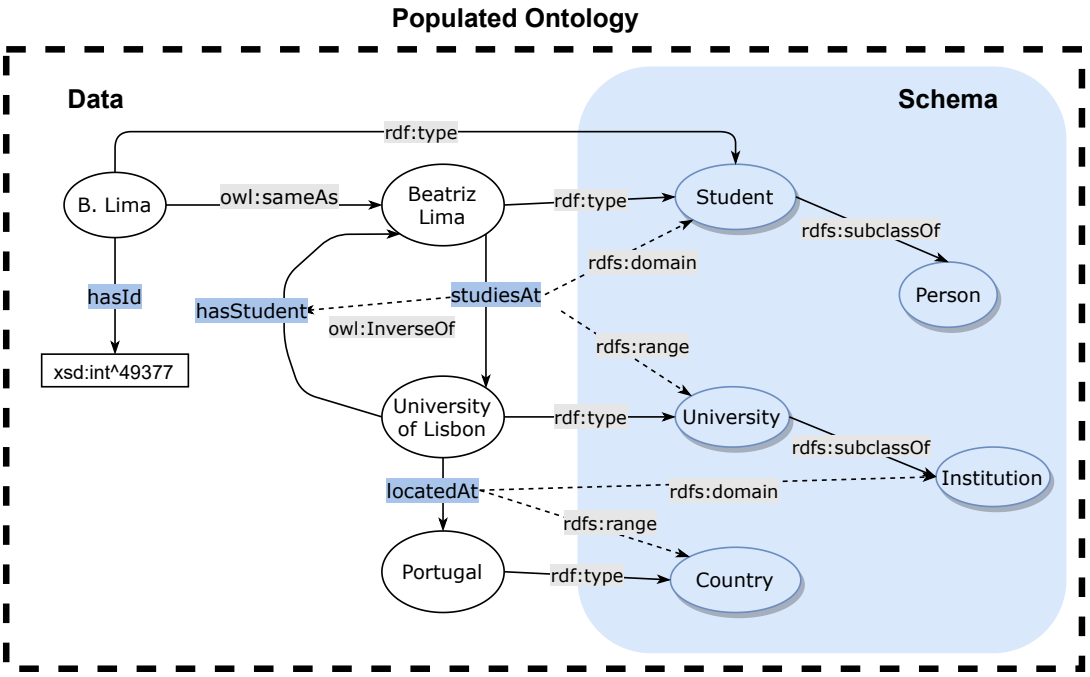


Figure 2.1: Example of an ontology populated with individuals. Coloured circles represent classes, while individuals are represented as uncoloured circles. Arrows represent properties.

2.3 Ontology Alignment

Ontology Alignment (OA) (or matching) is the process of producing entity mappings between ontologies of the same or intersecting domains. A *mapping* or *correspondence* is defined as the form $< e_1, e_2, r >$ where e_1 and e_2 are ontology entities from two related ontologies and r is the semantic relation that connects the concepts (e.g. equivalence ($\equiv$), subsumption ($\geq$ or $\leq$), disjointness ($\perp$), etc.) [52].

Alignments can be produced manually by humans, or automatically by *ontology alignment systems*, whose methodology typically follows the same general steps [17]. The system takes as input the source and target ontologies to be aligned and may also require some parameter specification and external resources. *Matchers* are algorithms with a central role in the alignment process and can vary greatly in the strategies used. The systems may run multiple matchers and output several alignments that need to be integrated into one final alignment. In those cases, filters or selectors may be employed in order to ensure the desired cardinality, by applying thresholds to discard less relevant mappings and by performing logical consistency checking [17].

The cardinality specifies how many entities can be assigned to each entity of the opposite ontology and the desired cardinality depends on the purpose of the application; for example, a cardinality of 1:1 specifies that each entity is only mapped to at most one other entity of the opposite ontology.

2.4 Complex Ontology Alignment

A Complex Ontology Alignment (COA) is an alignment that includes mappings with at least one complex entity.

While simple entities consist of a single ontology class, property or individual, with a unique identifier (e.g. $O1 : Teacher$), complex entities are expressions that involve not only single entities but also logical operators (e.g. $O2 : Educator$ or $O2 : Instructor$), restrictions on cardinality, type, range, value and more (e.g. $O2 : teaches\ some\ O2 : Student$) or transformation functions (e.g. $weight \times O2 : grade$). In the same line of thought, some examples of complex mappings are:

$$O1 : Teacher \leq O2 : Educator \text{ or } O2 : Instructor$$

$$O1 : Teacher \equiv O2 : teaches \text{ some } O2 : Student$$

Given a simple entity *s* and a complex entity *c*, complex mappings can assume the form *c:s* or *s:c*, depending on whether the complex entity belongs to the source or target ontologies, respectively. Complex mappings between two complex entities are denoted as *c:c* [52]. Another standard notation for complex mappings is *1:n*, *m:1* and *m:n*, which specify if the source and target entities are composed of a single (1) or several ontology entities (n, m) [52]. However, this notation is misleading, as some complex entities comprise only a single ontology entity (e.g. InverseOf($O1 : theaches$)).

Efficiently finding complex mappings is much a more challenging task than finding simple ones. While one can only find as many simple mappings as those resulting from the pairwise combination between the single entities of both ontologies, in complex matching, each mapping could involve many single entities and nested operators, leaving us with a boundless search space [38].

2.4.1 Complex patterns

Some of the most common patterns comprised in complex ontology alignments have been thoroughly described in the literature [30, 38, 41, 62]. They comprise several types of class and property expressions.

A class expression may be defined as an abstract class that groups the individuals that hold a certain relation (object or data property) to some restrict group of individuals or values (Table 2.1). Additional logic operators such as *and*, *or*, *not* may combine several class expressions.

A property expression may be defined as an abstract property obtained by restricting attributes such as domain and range of single properties to a specific group of individuals or values (Table 2.2). Operators such as *and*, *or*, *not*, and *compose* may combine several property expressions.

2.4.2 Representation format

There are several representation formats for complex mappings [52]. The Alignment API [16] provides a standard way to represent alignments, enabling them to be easily exchangeable among applications.

Table 2.1: Description of class expressions that compose common complex patterns.

Expression	**Definition**	**Example**
Object Property Range restriction	Constrains the range of an object property to a specific type of individuals.	$memberOf\ some\ Committee$
Object Property Cardinality restriction	Restricts the object property to be related to a specific number of individuals.	$min\ 1\ acceptedBy$
Data Property Value restriction	Restricts a data property to a specific literal value.	$hasTopic\ value\ "Science"$
Data Property Type restriction	Restricts a data property to a specific type of literal values.	$hasID\ some\ xsd : integer$

Table 2.2: Description of property expressions that compose common complex patterns.

Pattern	**Definition**	**Example**
Inverse Object Property	Defines an abstract property analogous to a given property, but with reversed domain and range.	$InverseOf\ contributes$
Object Property Domain restriction	Restricts the domain of a given object property to a certain type of individuals.	$writePaper \wedge domain(CoAuthor)$
Object Property Range restriction	Restricts the range of a given object property to a certain type of individuals.	$contributes \wedge range(Review)$
Data Property Domain restriction	Restricts the domain of a given data property to a certain type of individuals.	$hasID \wedge domain(ConferenceProceedings)$

The Expressive and Declarative Ontology Alignment Language (EDOAL) [1][10] was developed as an extension of the Alignment API to accommodate the representation of complex expressions in ontology alignments. EDOAL has been widely accepted by the community and it's currently employed in OAEI's Complex Track datasets [52].

Although heavily inspired by description logic and OWL restrictions, conjunctions, and disjunctions, it is an independent language, which allows it to also represent alignments of heterogeneous and weak representations such as thesauri and relational databases [17]. While some complex expressions could be covered by conventional OWL expressions, others such as the concatenation of property values (e.g., the values of *hasFirstName* and *hasLastName* in one ontology corresponding to value of *hasName* in another ontology), value restrictions with comparators (e.g. *hasAge* comparator:greater-than "18") and transformation functions (e.g. converting prices in one currency to another) go beyond OWL Semantics.

EDOAL features four types of expressions: constructions, restrictions, transformations and linkkeys. Analogously to the previously discussed elements that compose complex entities (Section 2.4.1), constructions amalgamate simple entities through algebraic operators, restrictions define new entities by narrowing the scope of existing entities, and transformations modify property values (although the suite of transformations supported by EDOAL is limited). Linkkeys establish equivalences between individuals of aligned entities.

As an example, take Figure 2.2 which represents the EDOAL representation of the mapping between class "$conference$: $Accepted\_contribution$" and the class defined by the union of "cmt : $hasDecision\ some\ cmt$: $Acceptance$" and "$min\ 1\ cmt$: $acceptedBy$".

All things considered, EDOAL supports the complex patterns presented in Section 2.4.1 and it is the representation format chosen for the implementation of this work.

2.5 Association Rule Mining

Association Rule Mining (ARM) [2] is a data mining technique that aims at finding structural frequent patterns. In this work, ARM is used to search for recurring relationships between entities of two ontologies among their common instances.

In the typical ARM process, the data is organised in a *transaction database*, where each transaction has a unique ID and contains a set of items, named an *itemset*. A k-itemset is one that contains k items [24]. The support computation is carried out for each k-itemset.

The *support* or frequency of an itemset is the percentage of transactions in the database that contain it. The support is used to find frequent patterns that are represented as *association rules*. Association rules are directional and comprise an antecedent and a consequent. *Strong* association rules satisfy both a minimum support and confidence threshold.

[1] https://moex.gitlabpages.inria.fr/alignapi/edoal.html

The *confidence* measure reflects the certainty of a rule and it is given by Equation 2.1, where A and B are itemsets, $sup(A \cup B)$ is the number of times that those itemsets were found together in the transaction database and $sup(A)$ is the support of itemset A.

$$Confidence(A \implies B) = \frac{sup(A \cup B)}{sup(A)} \tag{2.1}$$

Lift is a metric for correlation analysis commonly used in rule selection [24]. Lift (Equation 2.2) measures whether the events of a random transaction "containing all items in the antecedent" and "containing all items in the consequent" are statistically independent. Rules with a lift higher than 1 reflect a significant association between the antecedent and consequent, therefore being the most interesting ones.

$$Lift(A \implies B) = \frac{sup(A \cup B)}{sup(A) \cdot sup(B)} \tag{2.2}$$

Some of the most commonly used methods for frequent pattern mining include the Apriori [1] and FP-growth [23] algorithms.

The Apriori algorithm cuts down the search space by considering only *frequent itemsets*, i.e. those that occur at least as frequently as a minimum predetermined threshold, and relying on the observation that if an itemset is not frequent, none of its supersets can be frequent as well. It can be further optimised with the use of hash-based techniques, partitioning, sampling and more.

The FP-growth algorithm adopts a "divide and conquer" strategy, where the transaction database of frequent itemsets is compressed into a *frequent pattern tree* (FP-tree), where nodes represent itemsets and edges represent associations between itemsets. The tree is then divided into conditional trees, each associated with one frequent itemset, and conditional trees are mined separately. Hence, it does not require candidate itemset generation, contrarily to the Apriori algorithm. The FP-growth algorithm is about an order of magnitude faster than the Apriori algorithm, but can be troublesome for large databases, given that the tree may not fit in memory. In such cases, partioning [12, 40] and parallelization [6, 31, 59] strategies may be employed.

Figure 2.2: EDOAL representation of an example mapping.

```
<map>
    <Cell>
      <entity1>
        <edoal:Class rdf:about="http://conference#Accepted_contribution"/>
      </entity1>
      <entity2>
        <edoal:Class>
          <edoal:or rdf:parseType="Collection">
            <edoal:AttributeDomainRestriction>
              <edoal:onAttribute>
                <edoal:Relation rdf:about="http://cmt#hasDecision"/>
              </edoal:onAttribute>
              <edoal:all>
                <edoal:Class rdf:about="http://cmt#Acceptance"/>
              </edoal:all>
            </edoal:AttributeDomainRestriction>
            <edoal:AttributeOccurenceRestriction>
              <edoal:onAttribute>
                <edoal:Relation rdf:about="http://cmt#acceptedBy"/>
              </edoal:onAttribute>
              <edoal:comparator rdf:resource="http://edoal#greater-than"/>
              <edoal:value>0</edoal:value>
            </edoal:AttributeOccurenceRestriction>
          </edoal:or>
        </edoal:Class>
      </entity2>
      <relation>=</relation>
      <measure rdf:datatype='http://xsd#float'>1.0</measure>
    </Cell>
  </map>
```

Chapter 3

State of the art

Research in the ontology alignment field is mainly focused on simple alignments. The complex ontology alignment scope is somewhat recent, and still faces many challenges concerning datasets, benchmarks, evaluation, visualisation and edition of the alignments [52].

3.1 Complex Ontology Alignment

The Ontology Alignment Evaluation Initiative (OAEI) [1] introduced the Complex Matching track in 2018 [46], which currently provides seven datasets, covering five different domains: Conference, Populated Conference, Hydrography, Geolink, Populated Geolink, Populated Enslaved and Taxon.

The OAEI 2020 results [37] for this track show that automated complex ontology alignment is still a remote prospect. Out of the twelve participating systems, only three (AMLC [20, 32], AROA [65, 61], CANARD [48, 53]) were able to present complex mappings and their performance was very modest in comparison with the results of simple matching tracks (F-measures of at most 60%). Moreover, their efforts towards the systematic evaluation of complex alignment have brought to the forefront the difficulties in providing an accurate and fair evaluation of complex ontology alignments [65].

3.1.1 Methods

Complex ontology alignment approaches can be divided into two categories, according to the type of information explored: lexical and instance-based methods.

Lexical approaches [38, 39, 21] rely on finding patterns in the labels of the ontology entities, such as when the label of one class in one ontology overlaps with the label of both a class and a property in the other ontology (e.g. class *ProgramCommitteeMember* versus object property *memberOf* and class *ProgramCommittee*). Consequently, they are limited to finding mappings between entities with lexically

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

similar labels. Although the search space can potentially be extended through the use of thesauri, this approach will still fail to capture many real-world mappings (e.g. if a broader scope property such as *partOf* had been used instead of *memberOf* in the preceding example).

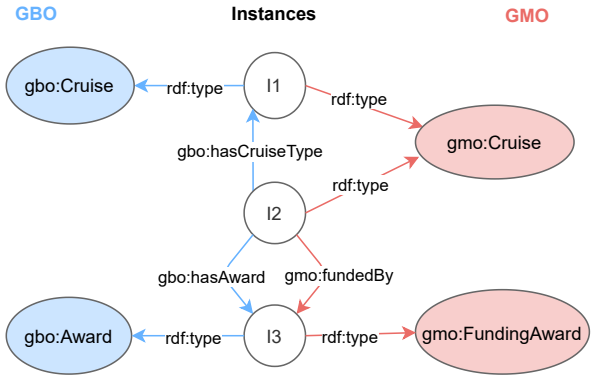
Instance-based approaches (e.g., [35, 58, 61]) explore data concerning individuals shared between the two ontologies. Their strategy is typically statistical or pattern mining-based. The intuition is that if there are recurring relationships between individuals and the classes and properties that describe them in opposite ontologies, those frequent patterns or correlations indicate the possibility of a semantic relation between the classes or properties. For example, if all individuals of class *O1: ProgramCommitteeMember* are connected to an individual of class *O2:ProgramCommittee* through *O2:memberOf*, a mapping such as $O1 : ProgramCommitteeMember \equiv O2 : memberOf$ some $O2 : ProgramCommittee$ can be inferred.

The mandatory use of instance data can be a limitation of instance-based approaches, but since data interoperability and integration are among the main applications of complex ontology alignment, instance data will be available more often than not. To obtain shared instance data, matching techniques can be used to map the individuals of two ontologies [37] with the accuracy being contingent on the lexical similarity between their entities' labels. Another limitation of these approaches is that their statistical nature makes them vulnerable to data sparseness, biases and errors, which can compromise the soundness of produced mappings [52].

3.1.2 Complex alignment systems

AMLC is a version of the alignment system AgreementMakerLight (AML) designed to perform complex ontology alignment [20, 19]. Its strategy consists of searching for lexical similarities between ontology entities according to a few predefined complex alignment patterns [38]. It allows for the optional use of an input simple alignment. This system does not rely on instance data for the alignment process, being the only OAEI participant capable of producing results in the non-populated datasets of the Complex track. However, it shows worse performance than the other two systems in some of the datasets where instance data is available since it cannot make use of this data.

AROA [65, 61] uses traditional ARM algorithms over a transaction database derived from the instance-level triples shared by two ontologies. As illustrated in Figure 3.1, the triples' *subjects* are represented as transaction IDs, being the *predicates* and *objects* represented as items. However, instead of having items as $property|instance$, since the goal is to establish mappings between the ontologies, the instance is substituted by its class or value ($property|class$ or $value$). The transaction database is then mined using the FP-Growth algorithm [23] and rules with patterns similar to those described in [38, 62] are selected to the final alignment.



Transaction DB	
Individual ID	**itemsets**
I1	rdf:type | gbo:Cruise rdf:type | gmo:Cruise
I2	gbo:hasCruiseType | gbo:Cruise rdf:type | gmo:Cruise gbo:hasAward | gbo:Award gmo:fundedBy | gmo:FundingAward
I3	rdf:type | gbo:Award rdf:type | gmo:FundingAward

Figure 3.1: Illustration of transaction database construction from instance-level triples as described in [62]. GBO and GMO are the ontologies that compose the Geolink dataset [63].

Following the example in Figure 3.1, some possible association rules would be:

- $rdf : type|gbo : Cruise \rightarrow rdf : type|gmo : Cruise$
- $gbo : hasCruiseType|gbo : Cruise \rightarrow rdf : type|gmo : Cruise$

translating into simple and complex class mappings, respectively. Some limitations to AROA include: it does not process *sameAs* relationships between individuals, and therefore does not produce results in the Populated Conference dataset; it requires prior knowledge of complex alignment patterns [61].

CANARD [48, 53] is a system that uses both a lexical and instance-based approaches. It relies on instance data and Competency Questions for Alignment (CQA) to discover complex mappings. CQAs "express the knowledge that an alignment should cover" [54]. In this approach, CQAs have the format of SPARQL queries over the source ontology, resulting in a set of instances. Those individuals are then matched to individuals annotated by the target ontology and their lexically similar surroundings are matched. CANARD's limitations include that it requires a set of domain-specific input questions to guide the matching process. They have been able to generate them automatically based on a set of patterns, as required for the OAEI participation [53]. It is also somewhat dependent on lexical similarity between ontology entities' labels. Additionally, CANARD's runtimes on OAEI 2020 were very long (up to 12 hours) which the authors justify by its dependence on the performance of the SPARQL endpoint among other factors [53].

3.1.3 Evaluation

While the evaluation of simple alignments is well established, the evaluation of complex alignments is still an open field of research.

The evaluation for simple mappings is primarily reference-based, consisting of an exact match with the reference, i.e. candidate mappings are considered correct if they are identical to a mapping in the

reference alignment, and incorrect otherwise. This approach is fairly adequate in the simple alignment realm, although some mappings could be considered semi-correct, such as a subsumption mapping between two classes that are in fact equivalent, or an equivalence mapping where one of the classes is a superclass of the correct class [14]. But in practice, such cases tend to be relatively rare and have little impact in evaluating matching systems on simple mappings.

However, this traditional approach is inadequate for complex mappings because the intricacy of the mappings and the unbound search space (due to the nesting of expressions) mean that cases where alignment systems predict complex mappings that approximate but do not exactly match those in the reference alignment are the norm rather than the exception. Moreover, two complex mappings can be syntactically different but semantically equivalent [52] (e.g. $Parent$ is equivalent to both $min\ 1\ hasChild$ and $hasChild\ some\ Child$). Considering these factors, the traditional evaluation approach is too unforgiving for complex mappings and does not accurately reflect their usefulness [65].

Manual evaluation is seldom used as it is a very time-consuming task, can introduce biases and inconsistencies to the evaluation [65], is not fully reproducible and may require domain expertise (especially for more complex mappings).

Zhou et al. [65] have overviewed existing evaluating approaches, dividing them into two families, depending on whether they require a reference alignment or not. They have discussed that to date, no evaluation approach that is simultaneously automated, comprehensive and able to accurately reflect the usefulness of complex alignments has been proposed. They argue that we are still far from achieving fully automatic evaluation and manual validation is still indispensable; in this scenario, an evaluation approach that reflected the expected human effort in validating a mapping, such as an edit-distance approach, would be the most suitable strategy for ontology integration tasks.

3.1.3.1 Evaluation with a reference alignment

Evaluation strategies dependent on a reference alignment compare the mappings of the evaluated alignment to those of a reference in order to compute Precision and Recall.

Zhou et al. [65] have detailed the general framework for this type of evaluation, which consists of: **anchoring**, mapping **comparison**, **scoring**, and **aggregation**. Additionally, they enumerate the challenges that should be addressed by a reference-based evaluation strategy (Table 3.1).

3.1.3.2 Evaluation without a reference alignment

Evaluation approaches that are not reference-dependent are an enticing solution as methods for generating reference alignments are still primarily manual, and, consequently, time-consuming and demanding of domain expertise. The non-reference-dependent evaluation approaches assess the overall quality of the alignment, being divided into two families, according to Zhou et al. [65]: one that aims to measure the logical soundness of an alignment [33, 44] and another which tests the fitness of the alignment for certain tasks [26, 27, 43].

Table 3.1: General framework for reference-based evaluation and challenges associated with each step, as described by Zhou et al. [65]

Step	Definition	Challenges (C)
Anchoring	Selection of the reference mapping(s) that should be compared to each candidate mapping	**C1:** avoid the necessity of a full pairwise comparison of reference and system mappings.
Comparison	Compute the relation between each pair of mappings.	**C2:** determine the relation between a candidate mapping and a reference mapping. **C3:** handle mapping decomposition (as two separate mappings can be equivalent to a single other mapping). **C4:** factor the mapping relation.
Scoring	Apply a scoring function to each relation identified in the comparison step	**C5:** accurately reflecting the quality/usefulness of each mapping.
Aggregation	Produce a final aggregated score which concerns the whole alignment.	**C6:** factor partially correct mappings. **C7:** factor cases of mapping decomposition. **C8:** handle the occurrence of (redundant) multiple candidate mappings that are implied by a single reference mapping.

Alignment quality metrics

Jiménez et al. [28] proposed the measurement of consistency and conservativity in order to assess the logical soundness of the aligments.

The consistency principle is related to the coherence of the alignment, predicated on the fact that it is desirable that the final alignment does not contain logical errors, in order for them to be compatible with certain applications. Meilicke et al. [33] proposed two metrics for assessing alignment logical coherence: one that takes into account the number of unsatisfiable classes, and another which is based on "the minimum number of mappings that must be removed to obtain a coherent merged ontology".

The conservativity principle states that alignments shouldn't contain "semantic relationships between any two entities that were originally from the same input ontology". Solimando et al. [44] proposed a metric that takes into account the number of violations to this principle.

These metrics do not reflect the alignment completeness or correctness, as an alignment can be fully

coherent and conservative but not encompass the most relevant mappings between two ontologies. Moreover, they are limited to evaluating alignments that are expressible in OWL DL, which is not always the case in the complex realm. The fact that the assessment of coherence requires reasoning strategies that can be computationally intensive when dealing with complex alignments is another major challenge.

Task-based evaluation

In practice, alignments are often built with the goal of being used in specific tasks or applications. Therefore, it is reasonable to evaluate their quality by assessing how well they perform in those tasks. Proposed tasks include ontology evolution, query answering [11, 43, 55] and thesaurus merging and data translation [27].

One of the downsides of these approaches is that, as the quality of the alignment is measured with respect to a particular task, the evaluation scope is narrowed. Additionally, Zhou et al. [65] discuss the challenges of generating generalizable quality metrics for the success of the task and, in the case of query answering–based strategies, rewriting SPARQL queries for more expressive mappings.

3.1.3.3 OAEI Complex track

As of the 2020 edition, the OAEI Complex track has established evaluation protocols for each of its datasets.

The Populated Conference task is automatically evaluated with Competency Questions for Alignment [45], assessing if the alignment can answer some basic queries. Precision is rendered by comparing the instances described by the source and target members of the mappings.

The evaluation of Taxon is task-oriented: first the quality of the generated alignments in terms of precision is manually assesed; second, there is a manual assessment of whether the alignment can answer SPARQL queries produced by a query rewriting system.

The evaluation of the Hydrography, Populated Geolink and Populated Enslaved tasks is based on an entity-based relaxed precision and recall approach, divided into three modalities: entity identification, relationship identification and full complex alignment identification. Performance is assessed using the relaxed precision and recall metrics [14], which penalise mappings that have correctly identified entities, but have failed to identify the correct relationship, according to a scale of similarity (Table 3.2).

Consider the following mappings from the $cmt-conference$ task of the OAEI (including reference and hypothetical candidate mappings) that will be used throughout this work as a running example:

- Reference mappings:

  (A) [$hasDecision$ some $Acceptance$] or [min 1 $acceptedBy$] = $Accepted\_contribution$
  (B) $ExternalReviewer$ = min 1 $inverseOf(invited\_by)$
  (C) $Reviewer$ or $ExternalReviewer$ = $Reviewer$

Table 3.2: Relationship identification task scores, which are attributed to a reference/candidate pair according to their relation similarity. Adapted from http://oaei.ontologymatching.org/2019/results/complex/hydrography/index.html

Score	Description	Ref. relation	Candidate relation
1.0	Correct relation	$=$ $>$ $<$	$=$ $>$ $<$
0.8	Candidate mapping returns less information, but is still correct	$<$ $=$	$=$ $>$
0.6	Candidate mapping returns more information, but is incorrect	$>$ $=$	$=$ $<$
0.3	Incorrect relation	$<$ $>$	$>$ $<$

- Candidate mappings:

  (A') $hasDecision$ some $Acceptance > Accepted\_contribution$
  (B') $ExternalReviewer =$ min 1 $invited\_by$
  (C') $Reviewer = Reviewer$

According to the steps identified in Table 3.1, the OAEI entity-based evaluation strategy would handle their evaluation as detailed bellow:

1. **Anchor** step: a manual preprocessing step converts reference and candidate mappings into a list of key-value pairs of related entities plus their mapping relation. The key is a source ontology entity (or combination of entities) belonging to the mappings and manually chosen to represent them (several mappings can share the same key if they have the same source entity). The value is the set of all remaining source and target ontology entities for the mapping(s) that have the key. The preprocessing of the running example mappings would result in the following key-value pairs:

   - Reference mappings:

     (A) $hasDecision : \{Accepted\_contribution, Acceptance, acceptedBy\}, =$
     (B) $ExternalReviewer : \{invited\_by\}, =$
     (C) $Reviewer : \{Reviewer, ExternalReviewer\}, =$

- Candidate mappings:

  (A') $hasDecision : \{Accepted\_contribution, Acceptance\}, >$
  (B') $ExternalReviewer : \{invited\_by\}, =$
  (C') $Reviewer : \{Reviewer\}, =$

2. **Correspondence comparison** step: each candidate mapping is compared with the reference mapping that has the same key-entity.

3. **Scoring** step: for each value-entities pair in the candidate mapping, entity-precision and entity-recall are computed against those pairs in the reference mapping. These score are then multiplied with a relation similarity score according to the criteria defined in Table 3.2.

4. **Aggregation** step: the final score of an alignment is the average of the entity scores. Applying this evaluation algorithm to the example above would result in the scores listed in Table 3.3.

Table 3.3: Scores obtained for the running examples under the OAEI entity-based evaluation strategy.

Alignment	**TP**	**FP**	**FN**	**Entity Precision**	**Entity Recall**	**Relation score**	**Relaxed Precision (%)**	**Relaxed Recall (%)**
A×A'	2	0	1	1	2/3	0.6	60	40
B×B'	1	0	0	1	1	1.0	100	100
C×C'	1	0	1	1	1/2	1.0	100	50
Final	-	-	-	-	-	-	86.7	63.3

3.2 Association Rule Mining in other Semantic Web-related tasks

The use of ARM for solving Semantic Web-related tasks is limited as it is not as scalable as other numeric Machine Learning technologies. While numeric-based ML approaches mainly target data in their graph representation, i.e. primarily plain RDF-based languages, symbol-based methods such as ARM easily incorporate more expressive languages such as OWL. ARM is also typically more interpretable than numeric-based ML approaches [9].

ARM as a pattern mining technique is suited for unsupervised tasks such as learning new rules or axioms given ontology instance and schema-level data. Moreover, it can be advantageous when deductive reasoning is impossible due to inconsistencies and incompleteness in ontologies. In this sense, ARM has been employed in the tasks of knowledge completion, ontology learning and learning disjointness axioms [9].

Knowledge completion is the task of finding assertions that are missing in knowledge bases. It is of particular interest to the fields of link prediction and Knowledge Graph development. Works in this area include systems that can automatically mine and generate semantically enriched rules from RDF data [3, 18, 29, 34].

Ontology learning is the process of constructing an ontology schema from RDF data. Völker et al. [56] have rendered that task by means of ARM.

Disjointness axioms are a way to represent the negative knowledge in a domain, but ontology authors often do not specify them. Völker et al. [57] showed that it is possible to find disjoint relations in the ontology schema by relying on correlation coefficients and negative association rules.

Chapter 4

Algorithms

4.1 Complex Ontology Alignment Algorithms

Traditional ARM requires a single database containing all the transactions to be mined, and employs an algorithm to find all frequent itemsets in the database, from which association rules are generated. Zhou *et al.* [62] demonstrated how a complex ontology alignment dataset can be transformed into a traditional ARM problem, and implemented this approach in AROA [61]. However, their approach relied on prior knowledge of the complex alignment patterns detailed by [38] to filter the ARM results and produce complex ontology mappings. Given prior knowledge of the complex alignment patterns, which translates to prior knowledge of the types of rules we want to find, then it is unnecessary and inefficient to use a "catch-all" ARM algorithm to perform an exhaustive search for frequent itemsets of all make-ups.

The strategy used in this work for efficient ARM-based Complex Ontology Alignment is two-fold. On the one hand, as complex alignment patterns are known *a priori*, the mining process can be carried out independently for each alignment pattern, concerning only the schema information that is relevant to the targeted pattern, thus reducing the search space. On the other hand, the incorporation of the ARM paradigm into a typical ontology alignment system [20] that parses ontology data into separate hash-table-based data structures for different types of data (e.g., hierarchical relations, lexical annotations) provides the basis for scalable alignment algorithms.

This approach will be mentioned as Pattern-Guided Association Rule Mining (PG-ARM) throughout this dissertation. Its pipeline is depicted in Figure 4.1, consisting of the following steps:

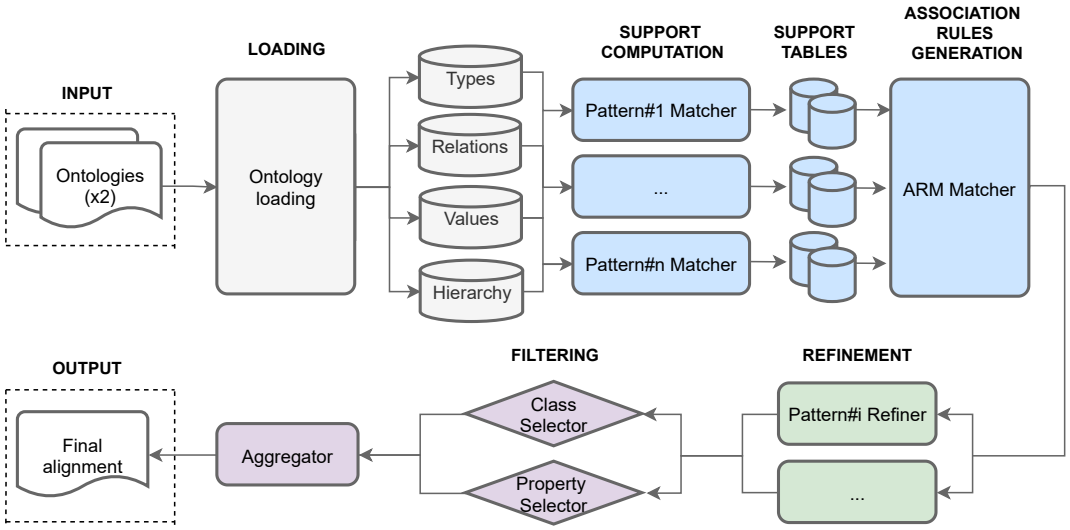


Figure 4.1: Complex ontology alignment pipeline based on pattern-guided association rule mining (PG-ARM). Grey indicates the external ontology loading system facilities, blue the matching algorithms, green the refinement algorithms and purple the filtering algorithms.

1. An initial ontology loading step renders the set of individuals shared by the two ontologies and efficient data structures (e.g. hash tables) which contain schema and instance-level data. These are:

   - **Individual Types**- holds the *rdf:type* relationships, i.e. class assignments, of individuals in the dataset;
   - **Individual Relations** - holds the relationships between individuals in the dataset and the object property that relates them. It also contains information on object properties' domains, ranges and other attributes;
   - **Individual Values** - holds the relationships between individuals and data values, and the data property that relates them;
   - **Hierarchy** - holds the hierarchical relationships between ontology classes and properties.

2. For each complex alignment pattern, an individual algorithm finds all mapping candidates (or item-sets) that match that pattern, by searching the appropriate data structures, and computes the support of the two entities (source and target) in the mapping candidate as well as the support of the mapping candidate itself (i.e., the fraction of shared individuals that have both the source and target entities).

3. A common ARM matching algorithm is invoked by each pattern matching algorithm to extract association rules, in the form of mappings, from the set of mapping candidates.

4. Refinement algorithms receive mappings generated by some of the pattern matching algorithms as input and refine those mappings, converting simple subsumption mappings into complex equivalence ones.

5. Filtering algorithms select which of the candidate mappings to include in the final alignment, excluding redundant mappings and conflicting mappings with lower confidence.

6. An aggregator combines mappings for the same entity into a single mapping using logical operators.

4.1.1 Matching algorithms

This work encompasses a total of nine matching algorithms, each corresponding to an alignment pattern. The alignment patterns chosen were based on EDOAL expressions [10]. A brief description on the patterns and the data structures required to find them are the following:

- **Class - Class**
  - **Type:** *s:s*
  - **Example:** $PaperAbstract \leq Abstract$
  - **Hash tables:** individual types; class hierarchy

- **Class - *someValues* restriction on Object Property**
  - **Type:** *s:c*
  - **Example:** $ProgramCommitteeMember \equiv memberOf$ some $ProgramCommittee$
  - **Hash tables:** individual types; individual relations; class hierarchy

- **Class - *cardinality* restriction on Object Property**
  - **Type:** *s:c*
  - **Example:** $AcceptedContribution \equiv$ min 1 $acceptedBy$
  - **Hash tables:** individual types; individual relations; class hierarchy
  - **Note**: limited to ”min 1” cardinality restrictions;

- **Class - *hasValue* restriction on Data Property**
  - **Type:** *s:c*
  - **Example:** $SciencePaper \equiv hasTopic$ value “Science”
  - **Hash tables:** individual types; individual values; class hierarchy

- **Class - *someValues* restriction on Data Property**
  - **Type:** *s:c*
  - **Example:** $PaperId \equiv hasID$ some $integer$
  - **Hash tables:** individual types; individual values; class hierarchy

- **Object Property - Object Property**
  - **Type:** *s:s*
  - **Example:** $hasAuthor \equiv writtenBy$
  - **Hash tables:** individual relations
- **Data Property - Data Property**
  - **Type:** *s:s*
  - **Example:** $email \equiv has\_an\_email$
  - **Hash tables:** individual values
- **Object Property - Data Property**
  - **Type:** *s:s*
  - **Example:** $acceptedBy \equiv isAccepted$
  - **Hash tables:** individual relations, individual values
- **Object Property - *InverseOf* Object Property**
  - **Type:** *s:c*
  - **Example:** $hasAuthor \equiv$ InverseOf $contributes$
  - **Hash tables:** individual relations

The matching algorithms share the same core structure that is detailed in Algorithm 1. The matching process can be divided into two steps: support computation and mapping generation.

4.1.1.1 Support computation

The algorithm iterates through the set of individuals shared by the two input ontologies and, for each individual, searches the hash table data structures containing the relevant data for the alignment pattern targeted by the matching algorithm. This approach bypasses the need to build a transaction database, by iterating directly through the alignment system's efficient data structures. A mapping candidate $< a, b >$ is generated if an individual is annotated to entities a and b (from opposite ontologies) that satisfy the pattern $A - B$. For each mapping candidate found, the support of source and target entities (*EntitySupport*), and the pair (*MappingSupport*) are incremented.

The ontology alignment format is directional, meaning that mappings are expected to be listed from a source ontology to a target ontology. Since the algorithms were not designed to be directional, i.e. they search for pattern $A - B$ regardless of A and B being from the source or target ontologies, both mappings $< a, b, rel >$ and $< b, a, rel >$ will be generated depending on which ontology comprises the entities a and b.

Hierarchical expansion of mapping candidates is performed for all patterns in which one of the entities is a single ontology class[1], i.e. mapping candidates are generated not only for the classes directly instanced by the individuals but also for all ancestors of those classes. This step allows for transitive closure, which is formally required in ontology alignments.

Take this example for the "Class - *someValues* restriction on Object Property" matching algorithm, which produces mappings of the type $< c_1, op_2 \wedge c_2, rel, score >$, where $c1$ is a class from the source ontology, *op* and $c2$ are an object property and class from the target ontology that compose the restriction on object property $op_2 \wedge c_2$. In a first step, support tables are produced by iterating through shared individuals and counting the *EntitySupport* for c_1 and $op_2 \wedge c_2$, and *MappingSupport* $c_1 \wedge op_2 \wedge c_2$:

- The *Individuals Types* hash table allows for counting how many individuals are of type c_1, thereby obtaining its *EntitySupport*. The combination of *Hierarchy* and *Individuals Types* tables allows us to extend the *EntitySupport* count to all of c_1's ancestors;

- The combination of the *Individual Relations* and *Individual Types* tables allows the count of support for the entity $op_2 \wedge c_2$, i.e. how frequently the object property op_2 ranges individuals of the type c_2;

- The *MappingSupport* is calculated concomitantly, attending to how frequently the two entities appear together ($c_1 \wedge op_2 \wedge c_2$).

The process is analogous for all mapping patterns, with merely the combinations of hash tables searched changing. For most of the pattern matching algorithms, the support computation can be done in a single iteration over the set of individuals shared by the two ontologies. The exception are the Property matching algorithms, that require an additional iteration over the set of properties found in the first iteration that connect the shared individuals.

The "Class - minimum *cardinality* restriction on Object Property" mappings are found by looking for sets of individuals of a given class in one ontology that have a relation through the same Object Property in the other ontology. This is similar to the "Class - *someValues* restriction on Object Property" except that no range specification is defined (the restriction is unqualified). Although cardinalities above one would be considered, at present the algorithm contemplates only cardinality 1.

Using traditional ARM terms, PG-ARM only searches for frequent 2-itemsets, as the goal is to find rules that establish correspondences between a source and a target entity. These entities can be simple or complex, according to the patterns described in 4.1.1. For this reason, PG-ARM algorithms are currently bound to finding *s:s* and *s:c* mappings.

[1] Includes "Class-Class", "Class - *someValues* restriction on Object Property", "Class - *cardinality* restriction on Object Property", "Class - *hasValue* restriction on Data Property" and "Class - *someValues* restriction on Data Property" matching algorithms.

4.1.1.2 Mapping generation

A common ARM core algorithm is invoked at this point, which first filters the mapping candidates by support and then by confidence or lift. The confidence and lift of the mapping are calculated according to Equations 4.1 and 4.2, respectively.

$$Confidence(A \implies B) = \frac{MappingSupport(A, B)}{EntitySupport(A)} \tag{4.1}$$

$$Lift(A \implies B) = \frac{MappingSupport(A, B)}{EntitySupport(A) \cdot EntitySupport(B)} \tag{4.2}$$

In this work, the minimum support threshold was set to 1% of the total number of individuals. The confidence and lift thresholds were set to 70% and 1.0, respectively. These thresholds were chosen empirically based on the Conference dataset results, in order to ensure adequate recall.

The support threshold chosen is very low which aligns with this application's purpose. Traditional ARM applications, such as market basket analysis, are generally interested in rules that describe prevailing patterns in the dataset, showing both ample dataset coverage and a high association between the antecedent and consequent; in this sense, they typically employ high support and confidence thresholds. However, the general goal of the ontology alignment task is to find rules for mapping as many individuals as possible, even those of low prevalence types, while simultaneously ensuring that the mappings are as precise as possible. In other words, even if a pattern only covers a few individuals in the dataset (low support), provided that all those individuals exhibit that pattern (high confidence) one would be interested in mapping the entities that compose the pattern.

As discussed in Section 2.5, a lift higher than 1 reflects a significant association between the antecedent and consequent, therefore being the chosen threshold for this correlation metric. Although negative correlation is extracted from lift values lower than 1, this work opted not to consider negative correlation given the nature of the matching task. The fact that two entities are negatively correlated does not always reflect that they are semantically opposite or disjunct (NOT and OR operators, respectively); the entities might describe separate groups of individuals that simply do not share the same vocabulary or domain. Moreover, this strategy underlies a Closed World assumption[2], which is not valid for ontologies, and could render many false mappings that would overburden further filtering steps. For these reasons, negative correlation was not explored in this work.

[2]The Closed World Assumption (CWA) is used in knowledge representation to define that if a statement is not currently known as true, then it is considered false. The opposite concept is the open-world assumption (OWA), stating that lack of knowledge does not imply falsity

Algorithm 1 ARM-based pattern matching algorithm

output: Set of mappings $< a, b, rel, score >$ or $< b, a, rel, score >$ where a and b are entities that compose a predefined alignment pattern $A - B$ (see Section 4.1.1), rel is the mapping relationship and $score$ is the mapping score.
input: Ontologies O_1 and O_2, their shared individuals S, hash tables H_A and H_B storing the data relevant to entity types A and B, and support and confidence thresholds min_s and min_c.
procedure Match(O_1, O_2, S, H_A, H_B, min_s, min_c)
    // Support count
    init: hash table $EntitySupport$, hash table $MappingSupport$, Set $Mappings$
    for i in S **do**
        for a in $H_A(i)$ **do**
            $EntitySupport(a)$ += 1;
            for b in $H_B(i)$ **do**
                if ($a \in O_1$ & $b \in O_2$) || ($a \in O_2$ & $b \in O_1$) **then**
                    $MappingSupport(a, b)$ += 1;
                end if
            end for
        end for
        for b in $H_B(i)$ **do**
            $EntitySupport(b)$ += 1;
        end for
    end for
    // Mapping generation
    for (a,b) in $MappingSupport$ **do**
        if $MappingSupport$(a,b) $> min_s$ **then**
            $conf_{a \Rightarrow b} \leftarrow MappingSupport(a, b)/EntitySupport(a)$
            $conf_{b \Rightarrow a} \leftarrow MappingSupport(a, b)/EntitySupport(b)$
            if $conf_{a \Rightarrow b} \geq min_c$ & $conf_{b \Rightarrow a} \geq min_c$ **then**
                $conf_{a,b} \leftarrow GeometricMean(conf_{a \Rightarrow b}, conf_{b \Rightarrow a})$
                if $a \in O_1$ **then**
                    $Mappings.add(a, b, \equiv, conf_{a,b})$;
                else
                    $Mappings.add(b, a, \equiv, conf_{a,b})$;
                end if
            else if $conf_{a \Rightarrow b} \geq min_c$ **then**
                if $a \in O_1$ **then**
                    $Mappings.add(a, b, \leq, conf_{a \Rightarrow b})$;

```
34:                 else
35:                     Mappings.add(b, a, ≥, conf_{a⇒b});
36:                 end if
37:             else if conf_{b⇒a} ≥ min_c then
38:                 if a ∈ O_1 then
39:                     Mappings.add(a, b, ≥, conf_{b⇒a});
40:                 else
41:                     Mappings.add(b, a, ≤, conf_{b⇒a});
42:                 end if
43:             end if
44:         end if
45:     end for
46:     return Mappings;
47: end procedure
```

4.1.2 Refinement algorithms

Refinement algorithms are used to capture alignment patterns contingent on simpler alignment patterns, which were found by previous matching algorithms (see Figure 4.1). The output is not a new alignment, but instead a modified alignment where the original mappings were replaced with the refined and semantically more accurate mappings whenever their confidence is higher.

These algorithms have a similar structure to that of matching algorithms (Algorithm 1) in the tasks of computing the support for relevant entities and generating mappings from association rules; the difference lies in the fact that they use an input alignment to reduce the search space for relevant entities.

For example, consider two simple entities of the same type, e_1 and e_2, and the true semantic relationship between them, $e_1 \equiv e_2 \cap X$, where X is any expression. The pattern $e_1 \leq e_2$ is logically implied by the true relation, and it is likely to be captured by the corresponding ARM-based matching algorithm. Thus, for any alignment pattern of the type $e_1 \equiv e_2 \cap X$, we can restrict the search space to the subsumption mappings found by the matching algorithms, rather than search through all of the shared individuals of the two ontologies. Furthermore, only shared individuals that are related to the broader entity $e2$ are considered in the search for X, as there is strong evidence that $e1$ implies $e2$.

The alignment patterns presently encompassed in the refinement algorithms, the data structures required to find them and the input mapping types they depend on are the following:

- **Object Property - Object Property + domain restriction**
  - **Type:** *s:c*
  - **Example:** $coWritePaper \equiv writePaper \sqcap$ domain $CoAuthor$
  - **Hash tables:** individual relations; property domains
  - **Input:** Object Property - Object Property subsumption mappings

- **Object Property - Object Property + range restriction**
  - **Type:** *s:c*
  - **Example:** $writeReview \equiv contributes \sqcap$ range $Review$
  - **Hash tables:** individual relations; property ranges
  - **Input:** Object Property - Object Property subsumption mappings

- **Data Property - Data Property + domain restriction**
  - **Type:** *s:c*
  - **Example:** $has\_an\_ISBN \equiv hasID \sqcap$ domain $ConferenceProceedings$
  - **Hash tables:** individual values; property domains
  - **Input:** Data Property - Data Property subsumption mappings

Many restriction expressions may be candidates to the final mapping, attending to their confidence scores; in such cases, they may be filtered or combined into one final expression (e.g. $e_1 \equiv e_2 \cap X \cap Y$ where X is a domain restriction, and Y is a range restriction).

4.1.3 Filtering algorithms

The outcome of the ARM matching and refinement algorithms is a complex ontology alignment that contains a substantial number of mappings, including multiple mappings per ontology entity.

In simple ontology alignments, the filtering task consists of finding the set of mappings that maximise the scores, while ensuring that each entity is only assigned once. This constitutes a fundamental combinatorial optimisation problem known as the Assignment Problem, which has a deterministic $\mathcal{O}(n^3)$ solution.

However, for complex alignments, things are not so straightforward, as complex expressions may involve several entities, and it is uncertain if entities should be restricted to integrating one complex entity at most, or if they should be allowed to engage in more than one assignment. Much like classes in ontologies can be declared equivalent to or subclasses of multiple class expressions, it is plausible for a class of one ontology to be equivalent to or a subclass of multiple class expressions in the other.

Nevertheless, it is necessary to remove semantically redundant mappings and select the best mapping(s) for each ontology entity, while excluding less reliable conflicting mappings as they may be artefacts of the dataset.

Considering these additional challenges to the Assignment Problem due to the nature of complex entities, the filtering algorithms developed in this work constitute a heuristic selection approach. The following sections delve into the general and pattern-specific criteria used in their design.

4.1.3.1 General criteria

The four pillars of my mapping selection heuristic are the relation, the confidence, whether the mapping is simple or complex and whether it is more generic or specific than related mappings.

First, attending that one of the main goals of complex ontology alignment is to enable data transformations to support data interoperability and integration, equivalence mappings are favoured over subsumption mappings, as the latter do not enable precise bidirectional transformations.

Second, mappings with 100% confidence are favoured over lower confidence ones, to ensure the transformations are valid for all available data.

Third, simple mappings are preferred over complex ones, in accordance with Occam's razor.

Finally, broader complex mappings are favoured over narrower ones to reduce the risk of overfitting the dataset and ensure the transformation is as encompassing possible.

In the context of an instance-based approach, one needs to consider dataset biases and that underrepresentation of ontology classes could lead to incorrect mappings. Considering the following mappings as an example,

- $Accepted\_contribution \equiv hasDecision$ some $Acceptance$
- $Camera\_ready\_contribution \equiv hasDecision$ some $Acceptance$

where $Accepted\_contribution > Camera\_ready\_contribution$. In a dataset where all instances of class $Accepted\_contribution$ are also from subclass $Camera\_ready\_contribution$, both these mappings would have the same confidence score, while only the first is accurate. In these situations, choosing more general mappings may be a safer option to avoid innacurate mappings.

4.1.3.2 Pattern-specific criteria

Some complex patterns may be more restricting than others. As explained in the previous section, the algorithms prefer broader complex mappings over more specific ones. We consider that a complex mapping is broader than another if:

- it has a broader simple entity (e.g. *Accepted_contribution* vs. its subclass *Camera_ready_contribution*);
- it has a broader restriction of the same type, i.e. a restriction of a broader property, with a broader range or domain, or with a broader cardinality;
- it has a broader complex pattern (e.g. *cardinality* restrictions are broader than *someValues* restrictions for the same property, which in turn are broader than *hasValue* restrictions).

Another pattern-specific criterion is that the mappings must be coherent with the ontology schemas; for example, domain and range restrictions must be compatible with the domain and range of the property declared in the ontology, i.e. one of its subclasses [3].

Take this analogous class mappings example, the first two containing $someValues$ restrictions and the last one containing a $cardinality$ restriction.

$$Accepted\_contribution \equiv acceptedBy \text{ some } ConferenceMember \tag{4.3}$$

$$Accepted\_contribution \equiv acceptedBy \text{ some } Administrator \tag{4.4}$$

$$Accepted\_contribution \equiv \text{min } 1 \ acceptedBy \tag{4.5}$$

The range specified in the ontology schema for the property $acceptedBy$ is class $Administrator$. Hence, all mappings restricting its range to a class that is not $Administrator$ or one of its subclasses is an incorrect mapping. Under this premise, the algorithm discards mapping 4.3. Comparing 4.4 to 4.5, the $cardinality$ restriction would be chosen as it is broader than the $someValues$ restriction. Moreover, the restriction in 4.4 is redundant, as it is the declared range of the property in the ontology.

The criteria used to filter property expressions is that mappings involving object properties are more reliable because incrementing the support of object property pairs requires them to relate the same two instances, while incrementing the support of data property pairs requires them to relate the same instance to a literal of the same datatype. Since this approach is purely instance-based and does not make use of string matching, it is not possible to compare literals. Additionally, mappings between properties of the same type (i.e. object-object and data-data) are preferred over object property-data property mappings.

4.1.4 Aggregation algorithm

If multiple mappings for the same entity persist following the semantic filtering step, an aggregation algorithm is applied. The algorithm combines all conflicting mappings, i.e. those that have a same common entity, into a single mapping using logical operators $\cap$ or $\cup$.

Conflicting mappings with 100% confidence imply that all individuals in the dataset related to the common entity are also covered by the conflicting entities, meaning that they invariably occur together. For this reason, the final mapping takes the form:

$$commonEntity \equiv conflictingEntity1 \cap ... \cap conflictingEntityN$$

On the other hand, mappings with confidence lower than 100% suggest that some individuals in the dataset related to the common entity are not related to some of the conflicting expressions. Thus we concatenate them using a union operator:

$$commonEntity \equiv conflictingEntity1 \cup ... \cup conflictingEntityN$$

[3] Note that if the class restricting the domain of a property is the same as the declared domain, then the mapping is redundant.

If by chance the $commonEntity$ is complex, the final generated mapping will be *c:c*. This aggregation step is the sole responsible for the generation of *c:c* mappings, since the pattern matching algorithms implemented in this work only cover *s:s* and *s:c* mapping patterns.

4.2 Complex Ontology Alignment Evaluation Algorithms

As discussed in Chapter 3, evaluation strategies non-dependent on reference alignments are able to assess the accuracy of data transformations to some extent, but are less comprehensive than evaluation based on a full reference alignment. For this reason, this work will only be focusing on reference-based evaluation strategies.

Although the OAEI's entity-based evaluation strategy is comprehensive, it is neither fully automated (as the transformation of complex mappings into mapped entities is done manually) nor entirely accurate (as it doesn't account for the semantic constructs in the complex mappings, only the entities). In order to address the former, I developed an algorithm that aims to automate the preprocessing step of this approach (Section 4.2.2) and, adressing the latter, I propose a novel element-overlap–based evaluation strategy (Section 4.2.1).

This work also comprises a fine grained manual evaluation, for which I have developed a scale based on the mappings correctness (Section 4.2.3).

4.2.1 Element-overlap–based evaluation

This work proposes a novel reference-based evaluation strategy that is fully automated, reproducible and open access. The element-overlap–based evaluation metric was inspired by Zhou *et al.* [65] and it aims to reflect the mapping correctness and quantify the expected effort to manually correct an alignment. It is not an edit-distance approach in the strict sense, as it reflects the similarity between the candidate and reference alignments, rather than the dissimilarity.

The detailed description of the element-overlap–based evaluation approach is presented in Algorithm 2. Given candidate and reference complex alignments, A_c and A_{ref}, the candidate and reference mappings are decomposed into lists of elements. Applying the preprocessing to the running example (see Section 3.1.3.3), we would obtain:

Reference mappings:

(A) $\{hasDecision$, $Acceptance$, or, min, 1, $accepted\_by$, =, $Accepted\_contribution\}$

(B) $\{ExternalReviewer$, =, min, 1, $InverseOf$, $invited\_by\}$

(C) $\{Reviewer$, or, $ExternalReviewer$, =, $Reviewer\}$

Candidate mappings:

(A') $\{hasDecision, Acceptance, >, Accepted\_contribution\}$

(B') $\{ExternalReviewer, =, \text{min}, 1, invited\_by\}$

(C') $\{Reviewer, =, Reviewer\}$

Note that the alignments are stored in data structures where each mapping is indexed by each of the ontology entities it contains, thus making the preprocessing step of trivial complexity. The element-overlap–based approach then follows the methodology described in Table 3.1:

1. **Anchor** step: for each mapping in the alignment, select the mappings from the reference sharing at least one entity from both ontologies.

2. **Correspondence comparison & Scoring** step: perform a syntactical comparison between the elements of each mapping to the selected reference mappings. Each mapping is scored by the maximum Weighted Jaccard score computed against the reference mappings. The weighted Jaccard score between two lists L_c and L_{ref} is given by:

$$\text{WJaccard}(L_c, L_{ref}) = \frac{\sum_{k \in L_c \cup L_{ref}} min(count(k, L_c), count(k, L_{ref}))}{\sum_{k \in L_c \cup L_{ref}} max(count(k, L_c), count(k, L_{ref}))}$$

   This is an adaptation of the traditional Jaccard score between sets, taking into account that the same element can occur multiple times in a list (as is the case in a complex mapping).

3. **Aggregation** step: Precision is computed as the average of the best scores obtained for each mapping in the candidate alignment. Recall is defined as the average of the best scores obtained for each mapping in the reference alignment. Although this approach enables reference mappings to be used for comparison multiple times, each one of them should be assigned to only one score (in this case, the highest). Following the running example, the resulting scores would be those listed in Table 4.1.

Table 4.1: Element-overlap scoring for the running example mappings.

Example	**Precision**	**Recall**
A×A'	3/8	3/8
B×B'	5/6	5/6
C×C'	3/5	3/5
Aggregate	60.3%	60.3%

Algorithm 2 Element-overlap–based evaluation algorithm

output: Precision and Recall metrics.
input: Candidate (A_c) and reference (A_{ref}) alignments.
function convert(A) ▷ Preprocessing
    init: HashTable $lists$
    for $mapping_i$ in A **do**
        for $element_j$ in $mapping_i$ **do**
            $lists$.add($mapping_i$,$element_j$)
        end for
    end for
end function

init: HashTable $lists_{ref}$ = convert(A_{ref}), $lists_c$ = convert(A_c), $Scores_{ref}$, $Scores_c$
init: double $Precision$ = 0
for $mapping_i$ in A_c **do** ▷ Anchoring
    init: $A_{r\_sources}$, $A_{r\_targets}$
    for $source\_entity_j \in mapping_i$ **do**
        $A_{r\_sources}$.addAll(A_{ref}.get($source\_entity_j$))
    end for
    for $target\_entity_j \in mapping_i$ **do**
        $A_{r\_targets}$.addAll(A_{ref}.get($target\_entity_j$))
    end for
    $A_{related}$ = $A_{r\_sources}$.retainAll($A_{r\_targets}$)
    for $mapping_j$ in $A_{related}$ **do** ▷ Comparison & Scoring
        sim = WJaccard($lists_c$.get($mapping_i$), $lists_{ref}$.get($mapping_j$))
        if $sim > Scores_{ref}$.get($mapping_j$) **then**
            $Scores_{ref}$.add($mapping_j$,sim)
        end if
        if $sim > Scores_c$.get($mapping_i$) **then**
            $Scores_c$.add($mapping_i$,sim)
        end if
    end for
    $Precision$ += $Scores_c$.get($mapping_i$) ▷ Aggregation
end for
$Precision$ /= A_c.size
init: double $Recall$ = 0
for $mapping_i$ in A_{ref} **do**
    $Recall$ += $Scores_{ref}$.get($mapping_i$)
end for
$Recall$ /= A_{ref}.size

4.2.2 Automation of the OAEI entity-based evaluation preprocessing

As explained in Section 3.1.3.3, the OAEI's entity-based evaluation strategy includes a manual preprocessing step whereby reference and candidate mappings are converted into key-value pairs of related entities plus the mapping relation. The fact this step is manual hinders scalability and reproducibility.

My proposed algorithm to automate the preprocessing step of this evaluation strategy was built as to emulate the manual process of identifying key-entities while operating under a set of rules that ensure an objective solution, to enable reproducibility. This set of rules was extracted through a process of reverse engineering, from the observation of the original manual files.

The algorithm first converts the reference alignment into key-value pairs under the following rules:

1. All mappings that have a single source entity will be identified by that entity as key, and have the set of target entities as value. If more than one mapping has the same key, the values will be merged.
2. All mappings that have multiple source entities will be identified by each of the source entities that is not already the key of a single-source mapping.
   (a) If there are multiple such source entities, the mapping will be decomposed into a key-value pair with each of those source entities as key, and the set of all target entities and all other source entities as value.
   (b) If there are no such source entities and the mapping contains exactly two source entities, it will be identified by the set of those two source entities as key.
   (c) If there are no such source entities and the mapping contains more than two entities, it will be identified by all pairwise combinations of source entities that are not keys of two-entity mappings.
      i. If there are multiple such pairs of source entities, the mapping will be decomposed into a key-value pair with each of those pairs as key.
      ii. If there is no such pair, the mapping will be identified by the set of all source entities.

Then, the candidate alignment is converted into key-value pairs using analogous rules, except that the reference alignment is used as anchor. For example, rule 2 becomes:

2'. All mappings that have multiple source entities will be identified by each of the source entities that is not the key of a single-source mapping **in the reference alignment**.

The same logic is applied to all rules, as the goal is to establish a parallel between the candidate alignment and the reference alignment so as to enable the evaluation of the former.

The remaining steps (beyond preprocessing) of the entity-based evaluation employed in this work were performed exactly as described in Section 3.1.3.3, as the OAEI organisers kindly shared the respective source code.

4.2.3 Manual evaluation scale

The manual evaluation strategy employed in this work classifies mappings according to a rating scale consisting of the following five categories with associated scores:

- Correct [1.0]: The mapping is formally correct (regardless of whether it is present in the reference alignment).
- Nearly correct [0.75]: Only minor changes would be necessary for the mapping to be correct (e.g. alter the mapping relation type or substitute a class for its sub- or super-class).
- Plausible [0.5]: The mapping seems sensible and no information in the ontologies or reference alignment contradicts it.
- Implausible [0.25]: The mapping does not seem sensible, and is likely an artefact derived from biases in the dataset, but no information in the ontologies or reference alignment contradicts it.
- False [0.0]: The mapping is contradictory to the reference alignment and/or ontologies.

These scores were used to compute a weighted precision which reflects how sensible and close to being true the mappings are, contemplating the context of the ontologies and instance data rather than assume the reference alignment as the only source of ground truth.

4.3 Implementation

This work's pattern-guided complex matching approach implementation[4], PG-ARM, consists of an extension of the complex version of the ontology alignment system AgreementMakerLight (AMLC) [19, 20]. This system was chosen for the following reasons: (1) it already has internal data structures required for complex ontology matching, including expressions and an implementation of the EDOAL alignment format [20]; (2) it can handle the incorporation of instance data [21]; and (3) its modular framework allows for an easy addition of new matchers and filters [19].

AML is based on the design principles of AgreementMaker [8, 7] while it is more qualified for the alignment of large ontologies given its added focus on efficiency. The core framework includes two modules: the ontology loading module and the ontology matching module.

The ontology loading module is concerned with the loading of the input ontology files and the construction of ontology objects, which are organised in efficient hash-based data structures. The ontology matching module is responsible for aligning the ontology objects by the means of multiple matchers and selectors.

[4]Code and data available at: https://github.com/AgreementMakerLight/AML-Project.git

PG-ARM does not make use of any of the existing matching or filtering algorithms implemented in AMLC, only making use of loading, exporting and data structure facilities. The results are then independent of this alignment system and could virtually be reproduced in any other.

4.3.1 Parsing of equivalent individuals

AML was not prepared to handle *owl:sameAs* relationships between individuals. However, the Populated Conference datasets don't encompass shared individuals *per se*, but rather each ontology has their own individuals, which are mapped to individuals of other ontologies (Figure 4.2). For this reason, I extended the ontology loading module to parse *owl:sameAs* relationships and, more importantly, extended the transitive closure so that (1) each individual could be linked to both ontologies' entities and (2) equivalent individuals would not be taken into account more than once in the support computation task. In this manner, the system was adapted to handle mapped instance data so that the algorithms could function ordinarily as if the individuals were indeed shared.

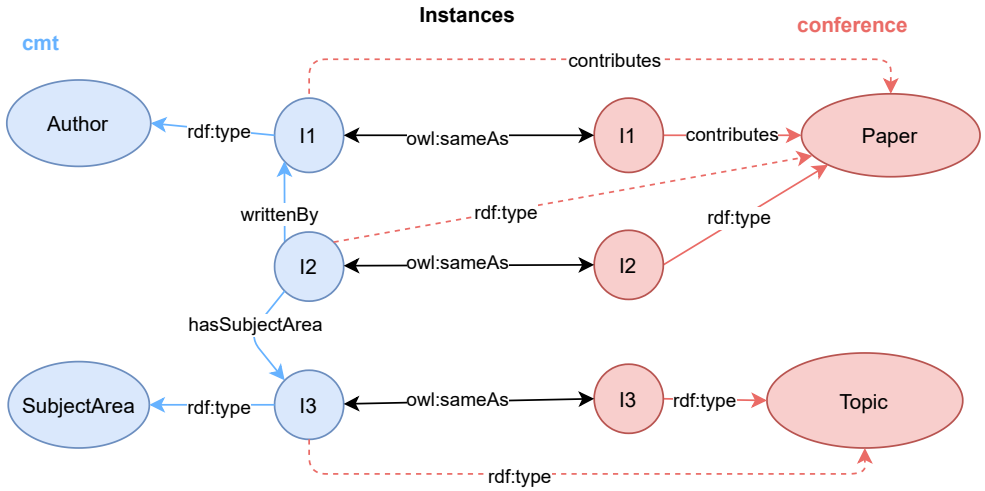


Figure 4.2: Example of mapped individuals in one of the Populated Conference alignments ($cmt - conference$). Dashed arrows represent the relationships that had to be added in transitive closure in order to simulate the shared instance scenario required in the matching algorithms.

4.3.2 Alignment visualisation

As complex EDOAL Alignments can be challenging to read and process by humans, a small tool [5] was developed for converting an EDOAL alignment into a human-friendly *csv* file, so that the analysis of

[5] Available at https://github.com/liseda-lab/EDOAL-2-CSV.git.

results could be eased.

The EDOAL alignment elements are divided into columns, having each line represent a mapping. The columns include the list of single entities involved in a complex entity expression, an external constructor (AND, OR, COMPOSE), mapping relationship and mapping score. It also includes columns with the type of entities involved (class or property) and whether the mapping is complex, for filtering purposes.

Additionally, the tool implements evaluation methods based on direct comparison of table cells in order to find exact matches between two alignments (usually reference and system alignment), but also the mappings that were missing in the system alignment, which facilitates an overview analysis of the results when assessing performance in terms of precision and recall, respectively.

Chapter 5

Evaluation

In order to evaluate the alignment algorithms proposed in this work, they were implemented in an existing alignment system, AMLC. The result alignments were evaluated under three modalities:

- Automated evaluation, under (5.2.1) a novel element-overlap–based approach and a (5.2.2) an implementation of the entity-based evaluation strategy employed in the OAEI, in order to evaluate the impact of filtering and aggregation and to compare this approach with state of the art systems;
- (5.2.3) Fine-grained manual evaluation to assess the performance on different patterns;
- (5.2.4) Run-time comparison with a traditional ARM algorithm FP-Growth.

5.1 Data

The datasets chosen for the evaluation of the proposed algorithms are two of the populated datasets available in the OAEI 2020 Complex track [1]. These datasets have quality reference alignments available and defined protocols for evaluation.

A general description of the datasets is provided in Table 5.1 and the summary of complex patterns present in their reference alignments can be found in Table 5.2.

The Populated Conference dataset [51] comprises five ontologies (*cmt, conference, confOf, edas* and *ekaw*). It is based on the OntoFarm dataset [60] and it covers the vocabulary on Academic conferences, articles, awards, etc. which is one that academics are familiar with, thus facilitating comprehension. Its reference alignments are provided by Thiéblin et al. [47].

The GeoLink dataset [63] is composed of two ontologies: the GeoLink Base Ontology (GBO) and the GeoLink Modular Ontology (GMO), which are inserted in the Geography domain. The reference alignment is curated by domain experts and the instance data are from real-worlds and can be found at the OAEI website.

[1] Available at http://oaei.ontologymatching.org/2020/complex/

Table 5.1: Description of datasets available in OAEI 2020. From left to right, columns describe: the number of ontologies that compose the dataset; number of simple (1:1) and complex (1:n, m:n) mappings; the number of individuals present in the dataset; the number of alignments between the dataset ontologies that have a reference alignment available.

Dataset	**Ontologies**	**1:1**	**1:n**	**m:n**	**Size (individuals)**	**Alignments**
Populated Conference (*v_100*)	5	111	86	98	137,311	20
Populated Geolink	2	19	5	43	22,301	1
Hydrography	4	113	69	15	10	4

Table 5.2: Occurrence of complex patterns (as described in 2.4.1) and logical operators in the reference alignments of the datasets used in this work's matching tasks. O.P: Object Property, D.P: Data Property.

	O.P. Range	O.P. Cardinality	D.P. Value	D.P. Type	Inverse OP	O.P. Domain	O.P. Range	D.P. Range	and	compose	or
Populated Conference	x	x			x	x	x		x	x	x
Populated Geolink		x				x	x	x	x	x	
Type of expression	**Class**				**Property**				**Operator**		

The Enslaved [64], Taxon [49] and Hydrography [46] datasets from the OAEI 2020 will not be used for the purpose of testing the matching algorithms.

There are unsolvable syntactical issues in the reference alignment of the Enslaved dataset, including ontology entities listed in the reference alignment which are not present in the ontologies or are present with a different namespace. No reference alignment is available for the Taxon dataset, thus precluding the reference-based evaluation.

The Hydrography dataset is not populated, which is incompatible with the instance-based algorithms proposed in this work; however, it is applicable for assessing the novel element-overlap–based evaluation strategy proposed, as it has reference alignments available. This dataset contains four source ontologies (Hydro3, HydrOntology_native, HydrOntology_translated and Cree) which are meant to be aligned to the target Surface Water Ontology (SWO). The source ontologies offer different challenges for the alignment task: Hydro3 is similar in both language and structure to the target, whereas HydrOntology, although similar in structure, is written in Spanish; Cree is very different from the target in both language and structure.

5.2 Complex Alignment Algorithm Evaluation

5.2.1 Element-overlap–based evaluation

The automated element-overlap evaluation results of this work's complex alignment approach for the populated Conference and Geolink datasets is presented in Table 5.3.

Table 5.3: Comparative **element-overlap–based** evaluation. The unfiltered, filtered and aggregated approaches correspond to PG-ARM alignments prior to filtering, after filtering, and after aggregation. Average and standard deviation reported for Conference. Size represents the number of mappings present in the alignment. The run times represent the time spent in the alignment process (excluding the ontology loading and evaluation).

Dataset	Approach	Precision (%)	Recall (%)	F-measure (%)	Size	Run time (s)
Conference	Unfiltered	2.8±0.5	52.3±10.2	5.3±0.9	326-1149	130±42
	Filtered	31.1±7.0	35.2±7.6	32.6±6.1	23-52	140±44
	Aggregated	41.6±9.8	33.9±7.3	**36.7±6.3**	16-38	142±46
	AMLC [20]	38.0±18.3	36.7±9.8	35.8±13.1	9-110	-
	CANARD [50]	23.5±13.1	43.2±8.3	28.7±11.0	34-172	-
	Reference	-	-	-	17-44	-
Geolink	Unfiltered	24.8	32.0	27.9	120	1
	Filtered	46.0	19.9	27.8	31	1
	Aggregated	63.6	16.1	25.7	18	1
	AMLC [20]	47.3	20.5	28.6	29	-
	AROA [61]	72.2	44.2	**54.8**	45	-
	CANARD [50]	53.5	32.6	40.5	41	-
	Reference	-	-	-	67	-

The performance prior to filtering, after filtering, and after aggregation was assessed. As expected, the filtering results in an increase in precision and a decrease in recall. It is noteworthy that aggregation allows for a substantial improvement on precision ($\sim$ 10-20%) with a considerably smaller loss in recall ($\sim$ 1-4%).

Comparing PG-ARM approach to other OAEI 2020 participating alignment systems, it achieves the highest F-measure in the Conference tasks, but performs worse than AROA and CANARD in Geolink. In Conference, it achieves the highest average precision, with a somewhat lower recall, whereas in Geolink it achieves the second highest precision, but considerably lower recall. These results are expected, since PG-ARM only includes matching algorithms for *s:s* and *s:c* mappings (with *c:c* only possible through

aggregation), and over 60% of the Geolink reference mappings are *c:c*.

The run times show that the system can function at real time and the filtering and aggregation steps are not significantly time consuming.

5.2.2 Entity-based evaluation

The entity-based evaluation results obtained for PG-ARM alignments prior to filtering, after filtering and after aggregation, as well as the evaluation of AMLC, CANARD and AROA alignments are presented in Table 5.4.

Table 5.4: Comparative **entity-based** evaluation. The unfiltered, filtered and aggregated approaches correspond to PG-ARM alignments prior to filtering, after filtering, and after aggregation. Average and standard deviation reported for Conference. Size represents the number of identified entities in the alignment.

Dataset	Approach	Relaxed Precision (%)	Relaxed Recall (%)	Relaxed F-measure (%)	Size
Conference	Unfiltered	4.7±1.3	46.8±12.0	8.5±2.0	28-82
	Filtered	39.4±8.2	39.4±10.7	38.8±7.7	14-45
	Aggregated	44.6±9.8	41.2±9.8	42.0±7.3	12-43
	AMLC [20]	48.7±14.2	38.1±12.0	**42.1±12.0**	8-53
	CANARD [50]	31.9±11.3	42.9±9.4	35.8±9.5	19-61
	Reference	-	-	-	16-44
Geolink	Unfiltered	37.4	31.0	33.9	36
	Filtered	67.7	19.7	30.5	25
	Aggregated	64.4	20.7	31.4	25
	AMLC [20]	50.3	23.1	31.7	28
	AROA [61]	87.9	45.8	**60.3**	38
	CANARD [50]	83.9	37.0	51.3	33
	Reference	-	-	-	57

As expected, the filtering resulted into a significant boost in precision (>20%), with some recall loss (>10%). As for aggregation, the results differ for each dataset and were also different from those obtained in the element-overlap evaluation; while it caused a precision increase in Conference, it had the opposite effect in Geolink, but both datasets showed an increase in recall.

It's difficult to interpret the aggregation effects on relaxed precision and recall, since the entity-based evaluation approach does not reflect these metrics in terms of the total number of mappings in the can-

didate and reference alignments, but rather the total number of entities identified. While the aggregation directly influences the number of mappings in the candidate alignment, its impact on the entities association is unpredictable.

The relaxed precision results imply that the aggregation somehow caused candidate alignment entities in Conference to be defined more correctly, but in Geolink the aggregation introduced wrong definitions to the identified alignment entities. Notice that the number of identified entities does not change considerably with the aggregation, which also justifies why the shifts in precision were not as high as in the element-overlap–based evaluation, where the total number of mappings was reduced by over 30%.

As for recall, both strategies showed an increase in this metric as a consequence of aggregation. One would not expect that more information about the reference would be gained by aggregating mappings; all this information would have been included in the filtered alignment as well, only decomposed. These results suggest that this evaluation strategy penalises decomposed mappings, a topic that will be further discussed in Section 5.3.

Table 5.4 also shows that AMLC and AROA achieved the highest F-measure scores in the Conference and Geolink tasks, respectively. Under the element-overlap–based evaluation, PG-ARM performed best in Conference, while the results were fairly close to AMLC.

Comparing the entity-based results to that of the element-overlap evaluation approach, the former are generally shifted upwards in terms of precision scores, while conserving the approaches' ranking for the most part.

5.2.3 Manual evaluation

The pattern-oriented summary of the manual evaluation of the *cmt-conference* task[2] is presented in Table 5.5. It yielded a global weighted precision of 75%, but revealed PG-ARM was unable to find mappings for some of the patterns present in the reference. Conversely, several mappings for patterns not present in the reference were found with high weighted precision ($\geq$78%).

Most patterns show similar weighted precision values (73.9-80.6%), but there are considerably lower values for "Class - *someValues* restriction on Object Property", with only 3 mappings out of 6 considered completely correct.

The discrepancy between the number of "Class - *cardinality* restriction on Object Property" mappings found by PG-ARM and those present in the reference was one of the most notorious. According to the manual evaluation, of the total 22 mappings found, 13 are considered completely correct and only 2 incorrect. Moreover, 2 out of the 5 mappings contemplated in the reference were found by the system and manually evaluated as correct or partially correct.

The higher weighted precision scores obtained in the manual evaluation in comparison with the automated evaluation is explained by the fact that many mappings that were not contemplated in the reference

[2]Full manual evaluation files available in Appendix (Chapter A)

Table 5.5: Pattern-oriented analysis of the results obtained in the $cmt - conference$ alignment using the filtered approach. N: number of mappings. Ref: reference alignment. W: weighted
* The total alignment size does not correspond to the sum of pattern occurrences as the same mapping may contain multiple patterns.

Pattern	**Ref. N**	**Result N**	**W.Precision (%)**
Class - Class	16	10	77.5
Class - *cardinality* restriction on Object Property	5	22	73.9
Class - *someValues* restriction on Object Property	4	6	58.3
Class - *hasValue* restriction on Data Property	-	-	-
Class - *someValues* restriction on Data Property	-	1	100
Object Property - Object Property	10	3	75.0
Data Property - Data Property	1	-	-
Object Property - Data Property	-	-	-
Object Property - *InverseOf* Object Property	2	-	-
Object Property - Object Property+range restriction	-	9	80.6
Object Property - Object Property+domain restriction	-	8	78.1
Data Property - Data Property+domain restriction	-	-	-
Total alignment*	35	51	75.0

were considered correct in the manual evaluation. These results show that the reference alignment is not exhaustive in all non-trivial correspondences that are valid between these two ontologies, suggesting that complex alignment references may be incomplete.

5.2.4 Comparison with traditional ARM approach

In order to assess the efficiency of the proposed ARM-based matching algorithms, a Java implementation of traditional ARM algorithm FP-growth [3] was used for comparison.

This implementation is a parallelised version of FP-Growth inspired by the work of Li et al. [31], which distributes the task of growing the FP-trees across several independent machines, thus being more scalable than traditional FP-Growth. The parallelised FP-Growth algorithm was shown to have virtually linear speedup in large mining tasks.

This implementation takes as input a transaction database in the form of a text file. For the purpose of this experiment, the transaction database was built from ontology triples, employing the same strategy as that of AROA (see Section 3.1.2). The transaction database is then mined in order to generate a set of

[3] https://spark.apache.org/docs/latest/mllib-frequent-pattern-mining.html

association rules, which could virtually be used to generate an alignment.

In order to compare the performance of PG-ARM with that of this FP-Growth Algorithm set up, it is assumed that the process ranging from the transaction database generation to association rule generation is the bottleneck of the matching task, and that time spent in the conversion of the association rules to an alignment format is negligible.

The chosen parameter values include minimum support of 1% of individuals and minimum confidence of 70%, same as this work's matching algorithms' settings. A total of 16 partitions were used in a Intel(R) Core(TM) i9-9900K CPU @ 3.60GHz, 32GB RAM machine.

The run times for the alignment process emulation using an FP-Growth, as well as the sizes of the transaction databases generated, are presented in Table 5.6. The run times shown correspond to the total time spent (1) generating the transaction database, (2) building the FP-Growth tree and (3) generating association rules.

Table 5.6: FP-Growth results. Columns from left to right show: the number of transactions generated in the process of aligning each pair of ontologies; the number of generated association rules and their text file size; Run times of (1) transaction database generation, (2) construction of the FP-tree, (3) association rule generation; PG-ARM runtimes shown for comparison. * Were not able to finish in a timely manner.

Dataset	**Transaction count**	**A. rules count**	**A. rules size**	**Run time (s)**			
				(1)	**(2)**	**(3)**	**PG-ARM**
gbo-gmo	10875	13884	3.7 MB	420	611	825	1
cmt-conference	26571	-	-	5641	2201	*	128
cmt-confOf	24730	-	-	2559	905	*	166
cmt-edas	26453	-	-	3050	929	*	144
cmt-ekaw	28178	-	-	5459	1457	*	106
conference-confOf	13460	-	>300 GB	2494	1291	>16h	388
conference-edas	26629	-	-	3874	1800	*	135
conference-ekaw	27298	-	-	4965	2222	*	88
confOf-edas	16641	-	-	2579	864	*	203
confOf-ekaw	15977	-	-	2934	1560	*	344
edas-ekaw	29485	-	-	5406	970	*	105

The alignment of the Conference ontologies was not possible as even the smallest dataset spent over 16 hours and produced files with sizes over 300GB, finally crashing down as the computer ran out of disk space. Table 5.6 shows that the bottleneck of this operation was the generation of association rules.

For the (1) and (2) tasks, the Conference run times were about one order of magnitude higher than Geolink's, which is as expected as the input ontology files are considerably larger.

As for the big discrepancy between Geolink and Conference run times in the association rule generation task (3), considering that the number of transactions in the Geolink database is not so different from that of the smallest Conference dataset, and attending to Figure 5.1, I hypothesise that the transaction size is the main responsible aspect. On the one hand, [25] have discussed that a greater number of items in transactions translates into larger frequent itemsets, which require more memory for storage and greater processing time to traverse them. On the other hand, there seems to be a large dispersion of transaction sizes in the Conference datasets, which constitutes a data skew that may influence the conditional FP-trees computation. This may be the cause for the significant increase in the run time and the memory errors. Hence, doing outlier removal on the transactions with disproportional amount of items might be a solution to consider.

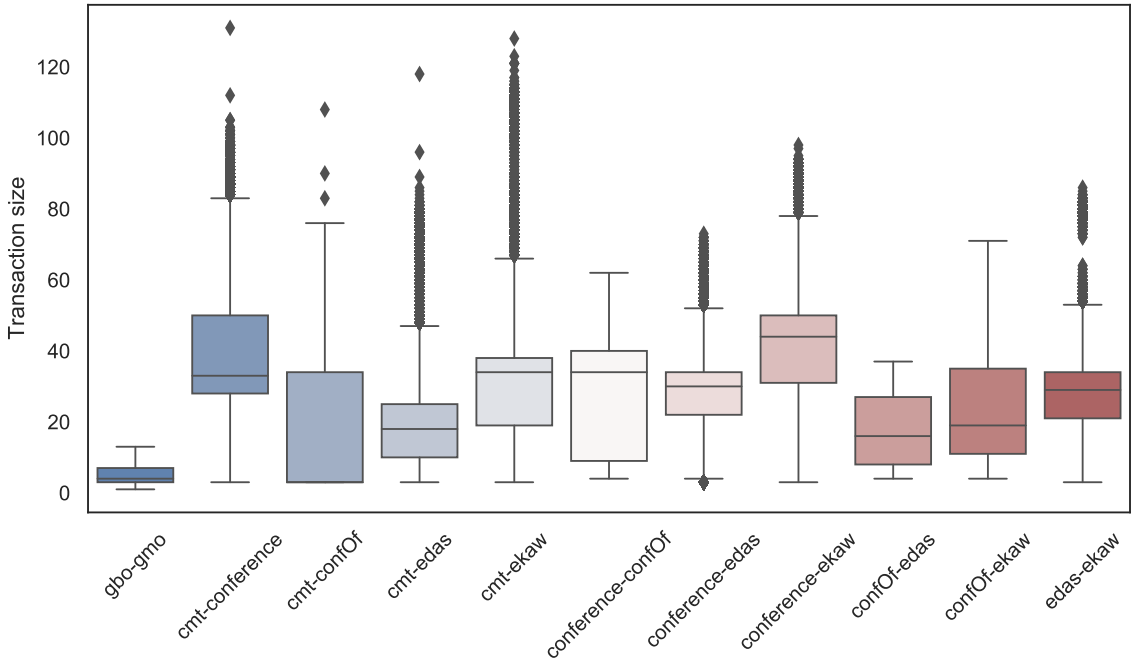


Figure 5.1: Distribution of transaction sizes in terms of number of items that compose them.

Comparing these results with PG-ARM run times, it appears that the FP-Growth implementation over a transaction database obtained from ontology triples takes significantly more time and resources than the pattern-guided algorithms presented in this work. However, it is important to note that the FP-Growth algorithm is not bound to 2-itemsets, as the PG-ARM algorithms currently are (as *c:c* patterns are not included), and thus works under a much vaster search space. Nevertheless, given that the generation of association rules wasn't even feasible for all datasets, these results show that a pattern-guided solution may be more reasonable, as long frequent patterns are neither very common nor necessary to the overall alignment of ontology entities.

5.2.5 Discussion

The performance assessment presented in this chapter places PG-ARM into a competitive position compared to other state-of-the-art complex ontology alignment systems.

The fine control over the alignment process and the refinement/filtering algorithms tailored to each alignment pattern proved to be an advantage of PG-ARM, as the filtering and aggregation rendered significant improvements in terms of precision, with low impact on recall. Additionally, the manual evaluation revealed that most of the patterns featured in PG-ARM are found with high precision.

On the other hand, this approach places more burden on the implementation, requiring an individual matching algorithm for each alignment pattern targeted. This work implements only algorithms spanning the most common patterns found in the Conference dataset, covering only *s:s* and *s:c* patterns, which is the main factor hampering PG-ARM's performance in Geolink.

Another advantage of PG-ARM is that it greatly reduces the ARM search space and enables an efficient exhaustive search, since one knows *a priori* the form of the rules to find. Employing a "catch-all" ARM algorithm such as FP-growth [23], that searches the whole space for rules of any and all types was show inefficient in this scenario (Section 5.2.4). Moreover, this reduction of the search space allows the hierarchical expansion of the rules – i.e., generate rules not only for the classes directly instanced by the individuals, but also for all ancestors of those classes – which is formally necessary in ontology matching, but in a traditional ARM approach leads to an explosion of the size of the transactions. The drawback of performing this expansion is that it leads to a much larger number of mappings, as evidenced by the unfiltered results (Table 5.3), thus demanding a filtering step.

5.3 Complex Evaluation Approach Comparison

Inadvertently, the development of complex alignment algorithms has made us stumble into the challenges of evaluating complex alignments, which are not fully addressed by existing evaluation strategies. This section is dedicated to comparing some of the evaluation strategies mentioned throughout this work, discussing the challenges they are able to tackle and those that remain neglected.

5.3.1 Element-overlap *vs.* entity-based evaluation

5.3.1.1 Comparative results

Table 5.7 summarises some of the results obtained using the element-overlap–based evaluation approach (**Element-overlap**) and the OAEI evaluation algorithm with the automated implementation of the preprocessing step (**OAEI auto.**), which were already presented in Section 5.2. Additionally, the OAEI evaluation algorithm with manual preprocessing (**OAEI man.**), as published in the OAEI website [4], is

[4] http://oaei.ontologymatching.org/2020/results/

also presented. For the Conference dataset, the OAEI evaluation was based on query answering, which is not comparable with the other two evaluation strategies, and, therefore, was omitted from the table. There are no OAEI manual results for PG-ARM, as it has not yet participated in the OAEI Complex track. Figure 5.2 provides a more intelligible visualisation of the performance of all alignment systems in the Conference dataset.

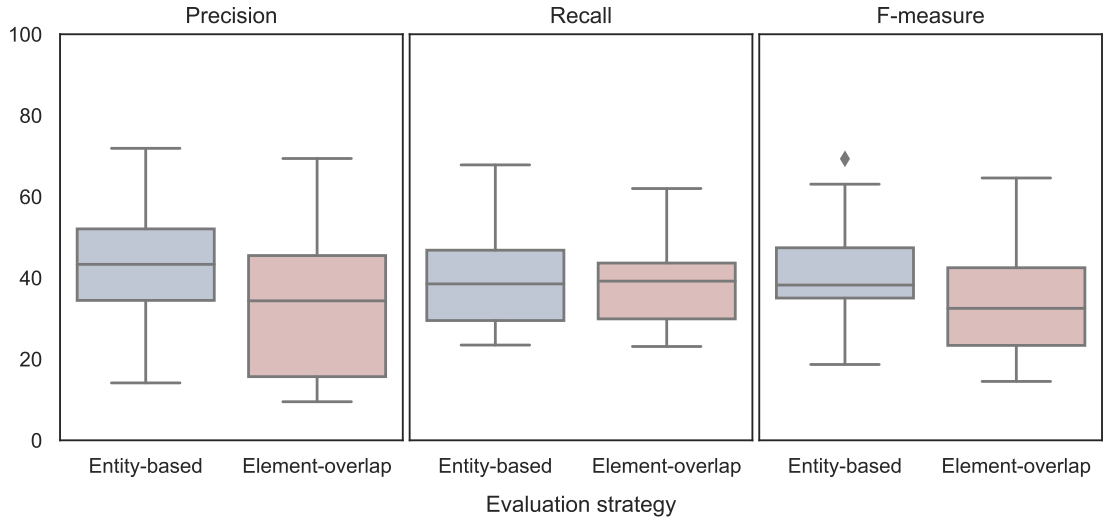


Figure 5.2: Conference Precision, Recall and F-measures scores according to the entity-based and element-overlap–based evaluation strategies, considering all alignment systems (AMLC, CANARD and PG-ARM).

The results show that the OAEI entity-based evaluation with automated preprocessing closely approximates the evaluation with manual preprocessing in most cases, with the only substantial difference being observed for CANARD in the Geolink dataset. However, the results of the two strategies were not the exact same, which suggests that the automated implementation did not replicate all the rules that went into the manual preprocessing of the alignments, although it provided a reasonable approximation. There were likely additional criteria of a different nature (e.g. favouring classes over properties as key-entities of mappings) which were not contemplated in this implementation.

It can be observed that the entity-based evaluation is consistently more generous in terms of precision than the element-overlap–based evaluation, while recall tends to be similar for both strategies. This can be attributed to the fact that the element-overlap–based approach factors both the ontology entities and the semantic constructs of the expressions in its scoring, whereas the entity-based evaluation factors only the entities. Since it is generally easier to automatically find related entities than to infer the exact semantic relations between them, matching systems would tend to score higher in precision under an entity-based evaluation.

Table 5.7: Evaluation of OAEI participating systems in the several complex datasets using the proposed element-overlap evaluation, the OAEI entity-based evaluation using the manual preprocessing step (OAEI man.) and this work's automated implementation (OAEI auto).

Alignment system	Evaluation strategy	Precision (%)	Recall (%)	F-measure (%)
Populated Conference				
AMLC	Element-overlap	38±18	37±10	36±13
	OAEI auto.	49±14	38±12	42±12
CANARD	Element-overlap	24±13	43±8	29±11
	OAEI auto.	32±11	43±9	36±10
PG-ARM	Element-overlap	42±10	34±7	37±6
	OAEI auto.	45±10	41±10	42±7
Populated Geolink				
AMLC	Element-overlap	47	21	29
	OAEI man.	50	23	32
	OAEI auto.	50	23	32
AROA	Element-overlap	72	44	55
	OAEI man.	87	46	60
	OAEI auto.	88	46	60
CANARD	Element-overlap	54	33	41
	OAEI man.	89	39	54
	OAEI auto.	84	37	51
PG-ARM	Element-overlap	64	16	26
	OAEI man.	-	-	-
	OAEI auto.	64	21	31
Hydrography				
AMLC	Element-overlap	43±15	8±10	12±14
	OAEI man.	48±17	7±8	12±13
	OAEI auto.	47±19	8±10	12±14

Through the study of the entity-based methodology, it has become apparent that both manual and automatic preprocessing are unnecessarily complex. While an entity-based evaluation simplifies the problem of evaluating nested and complex mappings, it seems arbitrary that each mapping would be represented by only key-entities, instead of all the entities in the mapping. Moreover, this strategy does not guarantee that a candidate mapping is compared to the best or most similar reference mapping available, as it may happen that their key entity is not the same. Further, the relaxed precision and recall scores are neither based on the number of mappings (as in a traditional evaluation and this work's element-overlap approach) nor based on the total number of mapped entities (as in a pure entity-based approach), but somewhere in between, making them hard to interpret or compare, as discussed in Section 5.2.2.

5.3.1.2 Addressing the evaluation challenges

This section aims to gauge the degree of which each of these two strategies address the challenges in evaluating complex alignments, as described in the literature [65]. Table 5.8 summarises this analysis.

Table 5.8: Challenges addressed by the element-overlap–based evaluation and the OAEI entity-based evaluation.

Challenge	Element-overlap	Entity-based
(C1) Avoid full pairwise	✓	✓
(C2) Relation between mappings	✓-	✓- -
(C3|C7) Mapping decomposition	✓- -	-
(C4) Mapping relation	✓	✓
(C5) Reflect usefulness	✓-	✓- -
(C6) Partially correct mappings	✓-	✓-
(C8) Redundant mappings	-	-

Starting with (C1), both strategies are able to avoid the necessity of a full pairwise comparison of reference and system mappings. The element-overlap–based evaluation does so by restricting the reference candidates to those that share at least one entity from both ontologies with the system mapping. The entity-based evaluation uses the entity identification step to assign each system mapping to the most suitable reference; reference source entities are identified first, and system entities are identified according to those.

As for (C2), Zhou et al. [65] mention that in order to determine the relation between two mappings, the comparison should encompass not only the singular entities but also the expressions in which they are listed for both mappings. Only the element-overlap–based approach addresses this issue, although with some limitations; for instance, the Jaccard similarity measure doesn't factor the order in which the elements appear in a mapping, and thus would not be able to distinguish between cases such as the following two hypothetical mappings that have the opposite meaning:

- ($Reviewer$ or $ExternalReviewer$) and (not $Author$) = $Reviewer$
- not ($Reviewer$ or $ExternalReviewer$) and $author$ = $Reviewer$

Concerning mapping decomposition (C3|C7), none of the strategies fully address this issue. This is supported by the results in Section 5.2.2, as mapping aggregation showed some effects (although minimal) on recall. An evaluation approach resilient to mapping decomposition would not have its recall score affected by the aggregation of mappings. Nevertheless, the element-overlap–based approach allows multiple candidate mappings to be compared against the same reference mapping thus providing an answer to this challenge to some extent.

Both evaluation strategies are able to factor the mapping relation (C4) and partially correct mappings (C6), although not completely in the case of the latter. I consider that the factoring of the mapping relation in the entity-based evaluation is richer than in the element-overlap–based approach, as it factors the semantic meaning of the relationship rather than if they simply match.

Assessing if the evaluation strategies successfully reflect the usefulness of mappings (C5) is a complex issue. From the human-validation point of view, identifying *which* entities are semantically related between two ontologies is more time-consuming than assessing *how* they are related. Nevertheless, there is still a cost to the latter, which should be factored into scoring the usefulness of a mapping. As an example, consider the two reference mappings (R1, R2) from the $conference - confOf$ task and the two corresponding hypothetical candidate mappings (S1, S2):

(R1) $Reviewed\_Contribution$ = min 1 InverseOf($reviews$)
(R2) $Reviewer$ = min 1 $reviews$
(S1) $Reviewed\_Contribution$ = min 1 $reviews$
(S2) $Reviewer$ = min 1 InverseOf($reviews$)

Under an entity-based evaluation, both candidate mappings would score 100% in precision and recall, since the presence of the *InverseOf* construct is invisible to this evaluation strategy. However, the reference states that these mappings are in fact wrong, as they have inverted the intended usage of the *reviews* property (i.e. its declared domain and range). With the element-overlap–based approach, on the other hand, the construct would be factored into the score, providing a more accurate measure of the usefulness of the mappings.

The impact of redundant mappings (C8) in the evaluation approaches was also studied. A manual removal of redundant mappings was carried out for CANARD's $conference - confOf$ alignment, as it contained a high number of these mappings.

In this context, redundant mappings are those that are semantically equivalent but syntactically distinct to some other mapping in the alignment. For instance, the following mappings are redundant since $has\_authors$ and has_a_review are the declared inverse properties of $contributes$ and $reviews$, respectively:

- [$contributes$ and domain($Reviewer$)] and [$reviews$ and domain($Review$)] = $reviews$
- [InverseOf($has\_authors$) and domain($Reviewer$)] and [InverseOf(has_a_review) and domain($Review$)] = $reviews$

Attending to Table 5.9, both evaluation strategies showed a gain in precision when the redundant mappings were removed. I posit that the the redundant mappings were given a low score in evaluation, thus their removal increased the final precision; I would expect the opposite effect when the redundant mappings are highly scored.

Table 5.9: CANARD's $conference - confOf$ alignment evaluation in the presence and absence of redundant mappings, according to the element-overlap–based evaluation approach and the automatic implementation of the OAEI evaluation (OAEI auto.).

Dataset	Evaluation strategy	Precision (%)	Recall (%)	F-measure (%)
With redundant mappings	Element-overlap	31	49	38
	OAEI auto.	40	53	46
Without redundant mappings	Element-overlap	39	48	43
	OAEI auto.	46	53	49

Moreover, the increase in recall after removing redundant mappings also evidences the element-overlap–based approach's lack of ability to handle redundant mappings. One would expect the recall to remain the same after removing redundant mappings, as they do not add any more information to the alignment. However, under the element-overlap–based approach, an evaluation artefact occurs: the (S1) mapping is evaluated using the reference mapping "$reviews$ and $contributes = reviewes$", but, since the second mapping (S2) does not include $contributes$ or $reviews$, it doesn't meet the minimum criteria that the evaluated and reference mappings should share at least one source and target entities. In this sense, (S2) would be evaluated against another reference mapping, "$Reviewer = reviewes$ min 1". This translates into an increase in recall, as an additional reference mapping was found for the redundant dataset.

In this manner, both evaluation strategies are affected by redundant mappings as they are not able to identify and process logically equivalent mappings. An OWL reasoner would be required for that task, which heavily increases the complexity of the evaluation. The complex alignments often include semantic constructs that aren't expressible in OWL and go beyond DL semantics, which compromises the decidability of the reasoning problem. Thus, while the element-overlap–based approach only provides a gross estimate of the usefulness of mappings, providing a significantly more accurate estimate in a scalable manner is not trivial.

All things considered, there is no challenge that the element-overlap–based evaluation approach addresses worse than the entity-based approach. For this reason, the former should be preferred over the latter.

5.3.2 Manual *vs.* element-overlap–based evaluation

Reference-based evaluation approaches are limited to the reference alignment quality. As discussed previously (Section 5.2.3), it is very difficult to provide all valid complex mappings between two ontologies which translates into incomplete references.

This section establishes a comparison between the mapping scores from the manual evaluation of $conference - confOf$ task to those attributed under the element-overlap–based evaluation strategy. Entity-based evaluation was not included in this analysis, as it doesn't attribute one score per mapping, but rather one score per entity, which makes the scores not quite comparable.

Figure 5.3 presents the distribution of scores according to both evaluation strategies. The results show that while the majority of mappings scored highly under the manual evaluation, there is a shift to very low score regions according to the element-overlap–based evaluation approach.

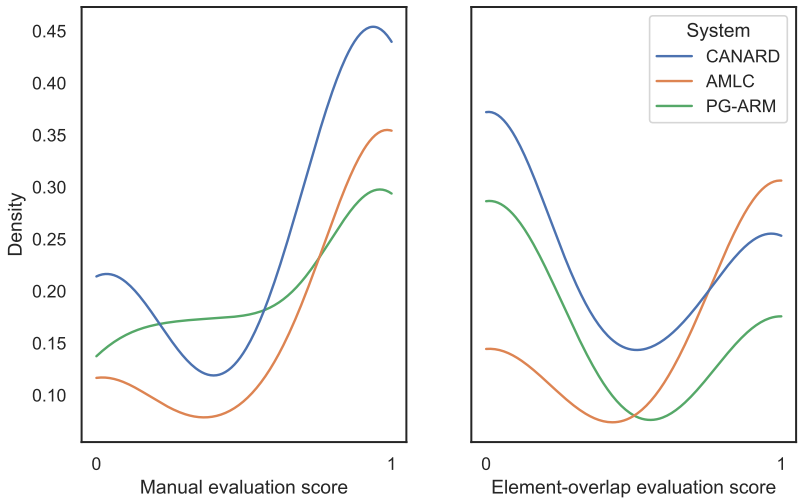


Figure 5.3: Distribution of mapping scores obtained from manual and element-overlap–based evaluation of conference-confOf.

Moreover, the correlation analysis of the scores assigned by the manual and element-overlap–based evaluation (Table 5.10) show that these two strategies are only moderately consensual on the semantic usefulness of the mappings, with the exception of the AMLC alignment, which rendered a very strong correlation between the manual and element-overlap scores attributed to each mapping.

Table 5.10: Pearson correlation of mapping scores attributed according to the manual and element-overlap–based evaluation approaches. The mappings belong to the $conference - confOf$ alignments of three alignment systems.

Alignment system	**Evaluation strategy**	Manual	Element-overlap
AMLC	Manual	1	0.949
	Element-overlap	-	1
CANARD	Manual	1	0.581
	Element-overlap	-	1
PG-ARM	Manual	1	0.672
	Element-overlap	-	1

These results corroborate those discussed in Section 5.2.3, with many correct mappings identified by the means of the manual evaluation not being contemplated in the reference alignment and thus being wrongly penalised in the reference-based evaluation. The complex ontology alignment field would definitely benefit from novel evaluation metrics that consider varying degrees of correctness beyond the reference alignment, while being fully automated.

Chapter 6

Conclusion

This work represents a paradigm shift in the Complex Ontology Alignment field, as instead of trying to fit COA to a traditional ARM setting as in previous work [65], the ARM process is designed around the problem of COA, taking advantage of the rich semantic information inherent to the dataset to cut down the search space and produce more sensible mappings to begin with, rather than a posteriori to process ARM results.

The pattern-oriented nature of the alignment algorithms allows for a fine control over the alignment and filtering process, as each algorithm is tailored to each pattern. Although this simultaneously limits the universe of patterns this approach can find, it seems that very long nested patterns are neither very common nor necessary to the overall alignment of ontology entities.

The element-overlap–based evaluation approach was proposed as an alternative to the OAEI evaluation strategies, addressing some of its shortcomings: (1) each mapping in the resulting alignment is compared only once, with the best fitting reference mapping; (2) it considers not only the entities present and mapping relation, but also the constructs; (3) it is fully automatic.

The comparative analysis of evaluation approaches showed that the entity-based evaluation employed in the OAEI is unnecessarily complex, and falls shorter of addressing the challenges identified for the evaluation of complex alignments [65] than the proposed element-overlap strategy. While this novel strategy knowingly sacrifices accuracy for scalability, a significant gain in accuracy is not easily achievable, due to complex mappings often falling outside DL semantics and thereby leading to undecidable reasoning problems.

6.1 Limitations

It is important to highlight some of the limitations of the alignment and evaluation algorithms proposed in this work. Concerning the former, their main limitation is that their pattern-oriented nature places more burden on the implementation, requiring an individual matching algorithm for each alignment pattern

targeted. Moreover, this work only contemplates algorithms for finding the most common *s:c* patterns found in the Conference dataset, excluding the following:

- cardinality restrictions on object properties beyond ”min 1”; this includes minimum restricitions with values other than 1, maximum and exact restrictions, as well as restrictions on data properties.
- *hasValue* restrictions on object properties.
- *allValues* restrictions on data/object properties.
- *c:c* patterns, including property chains (although some *c:c* mappings may be outputted by as a consequence of the aggregation process).

Additionally, the alignment algorithms require a populated dataset with shared instances to be available. Nevertheless, given that data interoperability and integration are one of the main applications of complex ontology alignment, instance data will often be available, and instance matching techniques can be used to map the individuals of two ontologies with typically high accuracy [37].

As for the proposed evaluation approaches, the element-overlap–based evaluation approach does not sufficiently address the challenges of assessing the relation between mappings, accounting for mapping decomposition and partially correct mappings. Being based on the Weighted Jaccard index, it only considers a ’bag’ of mapping elements, not being able to capture the order in which the elements appear. Additionally, it does not address redundant mappings at all.

6.2 Future work

This work paved the way for further studies, developments and optimisations. Regarding the evaluation approaches, one possible path is to perform the semantic comparison of mappings using simple rule-based approaches, in an attempt to provide a more accurate evaluation than that obtained with the proposed element-overlap–based strategy, without sacrificing scalability. Additionally, it could be interesting to incorporate more complex similarity metrics than the Weighted Jaccard index.

Going forward, it could be worthwhile to explore other datasets of linked data, making use of instance matching strategies for those that don’t encompass shared instance data. This is the case of OAEI’s Hydrography datasets, which differ in levels of complexity and are very rich in terms of cardinality restrictions and other complex patterns.

As for the alignment task, the development of algorithms to cover the patterns mentioned in 6.1 would increase PG-ARM’s scope with minimal decrease in run time performance. It would also be interesting to combine this purely instance-based strategy with a lexical approach, specially in filtering steps, which could most certainly enhance the system’s performance in trivial cases.

References

[1] R. Agrawal and R. Srikant. Fast Algorithms for Mining Association Rules in Large Databases. In *VLDB'94, Proceedings of 20th International Conference on Very Large Data Bases, September 12-15, 1994, Santiago de Chile, Chile*, pages 487–499. Morgan Kaufmann, 1994. 11

[2] R. Agrawal, T. Imielinski, and A. N. Swami. Mining Association Rules between Sets of Items in Large Databases. In P. Buneman and S. Jajodia, editors, *Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data, Washington, May 26-28, 1993, DC, USA*, pages 207–216. ACM Press, 1993. 10

[3] M. Barati, Q. Bai, and Q. Liu. Mining semantic association rules from RDF data. *Knowledge-Based Systems*, 133:183–196, 2017. 21

[4] T. Berners-Lee, J. Hendler, and O. Lassila. The Semantic Web. *Scientific American*, 284(5):34–43, 2001. 1, 5

[5] C. Bizer, T. Heath, K. Idehen, and T. Berners-Lee. Linked data on the web (LDOW2008). In *Proceedings of the 17th International Conference on World Wide Web, April 21-25, 2008, Beijing, China*, pages 1265–1266. ACM, 2008. 5

[6] D. W. Cheung, J. Han, V. T. Y. Ng, A. W. Fu, and Y. Fu. A Fast Distributed Algorithm for Mining Association Rules. In *Proceedings of the Fourth International Conference on Parallel and Distributed Information Systems, December 18-20, 1996, Miami Beach, Florida, USA*, pages 31–42. IEEE Computer Society, 1996. 11

[7] I. F. Cruz, F. P. Antonelli, and C. Stroe. AgreementMaker: Efficient Matching for Large Real-World Schemas and Ontologies. *Proc. VLDB Endow.*, 2(2):1586–1589, 2009. 38

[8] I. F. Cruz, C. Stroe, F. Caimi, A. Fabiani, C. Pesquita, F. M. Couto, and M. Palmonari. Using AgreementMaker to align ontologies for OAEI 2011. In *Proceedings of the 6th International Workshop on Ontology Matching, October 24, 2011, Bonn, Germany*, volume 814 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2011. 38

[9] C. D'Amato. Machine Learning for the Semantic Web: Lessons learnt and next research directions. *Semantic Web*, 11(1):195–203, 2020. 20

[10] J. David, J. Euzenat, F. Scharffe, and C. Trojahn. The alignment API 4.0. *Semantic Web*, 2(1):3–10, 2011. 10, 25

[11] J. David, J. Euzenat, P. Genevès, and N. Layaïda. Evaluation of Query Transformations without Data: Short paper. In *Companion of the The Web Conference 2018 on The Web Conference (WWW), April 23-27, 2018, Lyon , France*, pages 1599–1602. ACM, 2018. 18

[12] Y. Djenouri, J. C. Lin, K. Nørvåg, and H. Ramampiaro. Highly Efficient Pattern Mining Based on Transaction Decomposition. In *35th IEEE International Conference on Data Engineering, ICDE 2019, April 8-11, 2019, Macao, China*, pages 1646–1649. IEEE, 2019. 11

[13] A. Doan and A. Y. Halevy. Semantic Integration Research in the Database Community: A Brief Survey. *AI Mag.*, 26(1):83–94, 2005. 3

[14] M. Ehrig and J. Euzenat. Relaxed Precision and Recall for Ontology Matching. In B. Ashpole, M. Ehrig, J. Euzenat, and H. Stuckenschmidt, editors, *Integrating Ontologies '05, Proceedings of the K-CAP 2005 Workshop on Integrating Ontologies, October 2, 2005, Banff, Canada*, volume 156 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2005. 16, 18

[15] L. Ehrlinger and W. Wöß. Towards a Definition of Knowledge Graphs. In *Joint Proceedings of the Posters and Demos Track of the 12th International Conference on Semantic Systems and the 1st International Workshop on Semantic Change & Evolving Semantics (SuCCESS'16) co-located with the 12th International Conference on Semantic Systems (SEMANTiCS 2016),September 12-15, 2016, Leipzig, Germany*, volume 1695 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2016. 5, 6

[16] J. Euzenat. An API for Ontology Alignment. In *The Semantic Web - ISWC 2004: Third International Semantic Web Conference, November 7-11, 2004, Hiroshima, Japan*, volume 3298 of *Lecture Notes in Computer Science*, pages 698–712. Springer, 2004. 8

[17] J. Euzenat and P. Shvaiko. *Ontology Matching, Second Edition*. Springer, 2013. 1, 6, 7, 10

[18] D. Faria, A. Schlicker, C. Pesquita, H. Bastos, A. E. N. Ferreira, M. Albrecht, and A. O. Falcão. Mining GO Annotations for Improving Annotation Consistency. *PLOS ONE*, 7(7):1–7, 2012. 21

[19] D. Faria, C. Pesquita, E. Santos, M. Palmonari, I. F. Cruz, and F. M. Couto. The AgreementMakerLight Ontology Matching System. In *On the Move to Meaningful Internet Systems: OTM 2013 Conferences - Confederated International Conferences: CoopIS, DOA-Trusted Cloud, and ODBASE 2013, September 9-13, 2013, Graz, Austria. Proceedings*, volume 8185 of *Lecture Notes in Computer Science*, pages 527–541. Springer, 2013. 14, 38

[20] D. Faria, C. Pesquita, B. S. Balasubramani, T. Tervo, D. Carriço, R. Garrilha, F. M. Couto, and I. F. Cruz. Results of AML participation in OAEI 2018. In *Proceedings of the 13th International Workshop on Ontology Matching co-located with the 17th International Semantic Web Conference, Monterey, CA, USA, October 8, 2018*, volume 2288 of *CEUR Workshop Proceedings*, pages 125–131. CEUR-WS.org, 2018. 13, 14, 23, 38, 43, 44

[21] D. Faria, C. Pesquita, T. Tervo, F. Couto, and I. Cruz. AML and AMLC Results for OAEI 2019. In *Proceedings of the 14th International Workshop on Ontology Matching co-located with the 18th International Semantic Web Conference (ISWC 2019), October 26, 2019, Auckland, New Zealand*, volume 2536 of *CEUR Workshop Proceedings*, pages 101–106. CEUR-WS.org, 2019. 13, 38

[22] W. Hall and K. O'Hara. Semantic Web. In R. A. Meyers, editor, *Encyclopedia of Complexity and Systems Science*, pages 8084–8104. Springer, 2009. 1, 5

[23] J. Han, J. Pei, and Y. Yin. Mining Frequent Patterns without Candidate Generation. In *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data, May 16-18, 2000, Dallas, Texas, USA*, pages 1–12. ACM, 2000. 11, 14, 49

[24] J. Han, M. Kamber, and J. Pei. *Data Mining: Concepts and Techniques*. The Morgan Kaufmann Series in Data Management Systems. Elsevier Science, 2011. ISBN 9780123814807. 10, 11

[25] J. Heaton. Comparing dataset characteristics that favor the Apriori, Eclat or FP-Growth frequent itemset mining algorithms. In *SoutheastCon 2016*, pages 1–7, 2016. 48

[26] L. Hollink, M. van Assem, S. Wang, A. Isaac, and G. Schreiber. Two Variations on Ontology Alignment Evaluation: Methodological Issues. In *The Semantic Web: Research and Applications, 5th European Semantic Web Conference (ESWC), June 1-5, 2008, Tenerife, Canary Islands, Spain*, volume 5021 of *Lecture Notes in Computer Science*, pages 388–401. Springer, 2008. 16

[27] A. Isaac, H. Matthezing, L. van der Meij, S. Schlobach, S. Wang, and C. Zinn. Putting Ontology Alignment in Context: Usage Scenarios, Deployment and Evaluation in a Library Case. In *The Semantic Web: Research and Applications, 5th European Semantic Web Conference (ESWC), June 1-5, 2008, Tenerife, Canary Islands, Spain*, Lecture Notes in Computer Science, pages 402–417. Springer, 2008. 16, 18

[28] E. Jiménez-Ruiz, B. C. Grau, I. Horrocks, and R. B. Llavori. Logic-based assessment of the compatibility of UMLS ontology sources. *Journal of Biomedical Semantics*, 2(S-1):S2, 2011. 17

[29] J. Kralj, A. Vavpetič, M. Dumontier, and N. Lavrač. Network Ranking Assisted Semantic Data Mining. In *Bioinformatics and Biomedical Engineering - 4th International Conference (IWBBIO), April 20-22, 2016, Granada, Spain*, volume 9656 of *Lecture Notes in Computer Science*, pages 752–764. Springer, 2016. 21

[30] A. Krisnadhi, P. Hitzler, and K. Janowicz. On the Capabilities and Limitations of OWL Regarding Typecasting and Ontology Design Pattern Views. In *Ontology Engineering - 12th International Experiences and Directions Workshop on OWL, OWLED 2015, co-located with ISWC 2015, October 9-10, 2015, Bethlehem, PA, USA*, volume 9557 of *Lecture Notes in Computer Science*, pages 105–116. Springer, 2016. 8

[31] H. Li, Y. Wang, D. Zhang, M. Zhang, and E. Y. Chang. Pfp: Parallel Fp-Growth for Query Recommendation. In *Proceedings of the 2008 ACM Conference on Recommender Systems (RecSys), October 23-25, 2008, Lausanne, Switzerland*, pages 107–114. ACM, 2008. 11, 46

[32] B. Lima, D. Faria, F. Couto, I. Cruz, and C. Pesquita. OAEI 2020 results for AML and AMLC. In *Proceedings of the 15th International Workshop on Ontology Matching co-located with the 19th International Semantic Web Conference (ISWC), November 2, 2020, Virtual conference (originally planned to be in Athens, Greece)*, volume 2788 of *CEUR Workshop Proceedings*, pages 154–160. CEUR-WS.org, 2020. 13

[33] C. Meilicke and H. Stuckenschmidt. Incoherence as a Basis for Measuring the Quality of Ontology Mappings. In *Proceedings of the 3rd International Workshop on Ontology Matching (OM-2008) Collocated with the 7th International Semantic Web Conference (ISWC-2008), October 26, 2008, Karlsruhe, Germany*, volume 431 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2008. 16, 17

[34] V. Nebot and R. Berlanga. Finding association rules in semantic web data. *Knowledge-Based Systems*, 25(1):51–62, 2012. 21

[35] R. Parundekar, C. A. Knoblock, and J. L. Ambite. Linking and Building Ontologies of Linked Data. In *The Semantic Web - ISWC 2010 - 9th International Semantic Web Conference, ISWC 2010, November 7-11, 2010, Shanghai, China*, volume 6496 of *Lecture Notes in Computer Science*, pages 598–614. Springer, 2010. 14

[36] C. Pesquita, D. Faria, E. Santos, and F. M. Couto. To repair or not to repair: reconciling correctness and coherence in ontology reference alignments. 1111:13–24, 2013. 1

[37] M. A. N. Pour, A. Algergawy, R. Amini, D. Faria, I. Fundulaki, I. Harrow, S. Hertling, E. Jiménez-Ruiz, C. Jonquet, N. Karam, A. Khiat, A. Laadhar, P. Lambrix, H. Li, Y. Li, P. Hitzler, H. Paulheim, C. Pesquita, T. Saveta, P. Shvaiko, A. Splendiani, É. Thiéblin, C. Trojahn, J. Vatascinová, B. Yaman, O. Zamazal, and L. Zhou. Results of the Ontology Alignment Evaluation Initiative 2020. In *Proceedings of the 15th International Workshop on Ontology Matching co-located with the 19th International Semantic Web Conference (ISWC 2020), November 2, 2020, Virtual conference (originally planned to be in Athens, Greece)*, volume 2788 of *CEUR Workshop Proceedings*, pages 92–138. CEUR-WS.org, 2020. 1, 13, 14, 58

[38] D. Ritze, C. Meilicke, O. Sváb-Zamazal, and H. Stuckenschmidt. A Pattern-based Ontology Matching Approach for Detecting Complex Correspondences. In *Proceedings of the 4th International Workshop on Ontology Matching (OM-2009) collocated with the 8th International Semantic Web Conference (ISWC-2009), October 25, 2009 Chantilly, USA*, CEUR. 8, 13, 14, 23

[39] D. Ritze, J. Völker, C. Meilicke, and O. Sváb-Zamazal. Linguistic analysis for complex ontology matching. In *Proceedings of the 5th International Workshop on Ontology Matching (OM-2010), November 7, 2010, Shanghai, China*, volume 689 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2010. 13

[40] A. Savasere, E. Omiecinski, and S. Navathe. An efficient algorithm for mining association rules in large databases. In *21st VLDB Conf.*, pages 432–444. Morgan Kaufmann, 1995. 11

[41] F. Scharffe. *Correspondence patterns representation*. PhD thesis, University of Innsbruck, 2009. 8

[42] A. P. Sheth and C. Ramakrishnan. Semantic (Web) technology in action: Ontology driven information systems for search, integration, and analysis. *IEEE Data Engineering Bulletin*, 26(4):40, 2003. 6

[43] A. Solimando, E. Jiménez-Ruiz, and C. Pinkel. Evaluating ontology alignment systems in query answering tasks. In *Proceedings of the ISWC 2014 Posters & Demonstrations Track a track within the 13th International Semantic Web Conference, ISWC 2014, October 21, 2014, Riva del Garda, Italy*, volume 1272 of *CEUR Workshop Proceedings*, pages 301–304. CEUR-WS.org, 2014. 16, 18

[44] A. Solimando, E. Jiménez-Ruiz, and G. Guerrini. Minimizing conservativity violations in ontology alignments: algorithms and evaluation. *Knowledge and Information Systems*, 51(3):775–819, 2017. 16, 17

[45] É. Thiéblin. Do competency questions for alignment help fostering complex correspondences? In *Proceedings of the EKAW Doctoral Consortium 2018 co-located with the 21st International Conference on Knowledge Engineering and Knowledge Management (EKAW), November 13, 2018, Nancy, France*, volume 2306 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2018. 18

[46] É. Thiéblin, M. Cheatham, C. Santos, O. Sváb-Zamazal, and L. Zhou. The First Version of the OAEI Complex Alignment Benchmark. In *Proceedings of the ISWC 2018 Posters & Demonstrations, Industry and Blue Sky Ideas Tracks co-located with 17th International Semantic Web Conference (ISWC 2018), October 8-12, 2018, Monterey, USA*, volume 2180 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2018. 13, 42

[47] É. Thiéblin, O. Haemmerlé, N. Hernandez, and C. Trojahn. Task-Oriented Complex Ontology Alignment: Two Alignment Evaluation Sets. In *The Semantic Web - 15th International Conference*

(ESWC), June 3-7, 2018, Heraklion, Crete, Greece, volume 10843 of *Lecture Notes in Computer Science*, pages 655–670. Springer, 2018. 41

[48] É. Thiéblin, O. Haemmerlé, and C. Trojahn. Complex matching based on competency questions for alignment: a first sketch. In *Proceedings of the 13th International Workshop on Ontology Matching co-located with the 17th International Semantic Web Conference, October 8, 2018, Monterey, CA, USA*, volume 2288 of *CEUR Workshop Proceedings*, pages 66–70. CEUR-WS.org, 2018. 13, 15

[49] É. Thiéblin, N. Hernandez, C. Roussey, and C. Trojahn. Cross-querying LOD data sets using complex alignments: an experiment using AgronomicTaxon, Agrovoc, DBpedia and TAXREF-LD. *International Journal of Metadata, Semantics and Ontologies*, 13(2):104–119, 2018. 42

[50] É. Thiéblin, O. Haemmerlé, and C. Trojahn. CANARD complex matching system: results of the 2019 OAEI evaluation campaign. In *Proceedings of the 14th International Workshop on Ontology Matching co-located with the 18th International Semantic Web Conference (ISWC), October 26, 2019, Auckland, New Zealand*, volume 2536 of *CEUR Workshop Proceedings*, pages 114–122. CEUR-WS.org, 2019. 43, 44

[51] É. Thiéblin, M. Cheatham, C. Trojahn, and O. Zamazal. A consensual dataset for complex ontology matching evaluation. *The Knowledge Engineering Review*, 35:e34, 2020. 41

[52] É. Thiéblin, O. Haemmerlé, N. Hernandez, and C. Trojahn. Survey on complex ontology matching. *Semantic Web*, 11(4):689–727, 2020. 3, 7, 8, 10, 13, 14, 16

[53] É. Thiéblin, O. Haemmerlé, and C. Trojahn. Results of CANARD in OAEI 2020. In *Proceedings of the 15th International Workshop on Ontology Matching co-located with the 19th International Semantic Web Conference (ISWC), November 2, 2020, Virtual conference (originally planned to be in Athens, Greece)*, volume 2788 of *CEUR Workshop Proceedings*, pages 176–180. CEUR-WS.org, 2020. 13, 15

[54] E. Thiéblin, O. Haemmerlé, and C. Trojahn. Generating Expressive Correspondences: An Approach Based on User Knowledge Needs and A-Box Relation Discovery. In J. Z. Pan, V. Tamma, C. D'Amato, K. Janowicz, B. Fu, A. Polleres, O. Seneviratne, and L. Kagal, editors, *The Semantic Web – ISWC 2020*, pages 565–583, Cham, 2020. Springer International Publishing. 15

[55] É. Thiéblin, O. Haemmerlé, and C. Trojahn. Automatic evaluation of complex alignments: An instance-based approach. *Semantic Web*, 12(5):767–787, 2021. 18

[56] J. Völker and M. Niepert. Statistical Schema Induction. In *The Semantic Web: Research and Applications*, volume 6643 of *Lecture Notes in Computer Science*, pages 124–138. Springer, 2011. 21

[57] J. Völker, D. Fleischhacker, and H. Stuckenschmidt. Automatic acquisition of class disjointness. *Journal of Web Semantics*, 35:124–139, 2015. 21

[58] B. Walshe, R. Brennan, and D. O'Sullivan. Bayes-ReCCE: A Bayesian Model for Detecting Restriction Class Correspondences in Linked Open Data Knowledge Bases. *International Journal on Semantic Web and Information Systems (IJSWIS)*, 12(2):25–52, 2016. 14

[59] M. J. Zaki, S. Parthasarathy, M. Ogihara, and W. Li. Parallel Algorithms for Discovery of Association Rules. *Data Mining and Knowledge Discovery*, 1(4):343–373, 1997. 11

[60] O. Zamazal and V. Svátek. The Ten-Year OntoFarm and its Fertilization within the Onto-Sphere. *Journal of Web Semantics*, 43:46–53, 2017. 41

[61] L. Zhou and P. Hitzler. AROA Results for 2020 OAEI. In *Proceedings of the 15th International Workshop on Ontology Matching co-located with the 19th International Semantic Web Conference (ISWC), November 2, 2020, Virtual conference, Athens, Greece*, volume 2788 of *CEUR Workshop Proceedings*, pages 161–167. CEUR-WS.org, 2020. 13, 14, 15, 23, 43, 44

[62] L. Zhou, M. Cheatham, and P. Hitzler. Towards association rule-based complex ontology alignment. In *Proceedings of the 9th Joint International Semantic Technology Conference (JIST), November 25-27, 2019, Hangzhou, China*, volume 12032 of *Lecture Notes in Computer Science*, pages 287–303. Springer, 2019. XI, 8, 14, 15, 23

[63] L. Zhou, M. Cheatham, A. Krisnadhi, and P. Hitzler. GeoLink Data Set: A Complex Alignment Benchmark from Real-world Ontology. *Data Intelligence*, 2(3):353–378, 2020. XI, 15, 41

[64] L. Zhou, C. Shimizu, P. Hitzler, A. M. Sheill, S. G. Estrecha, C. Foley, D. Tarr, and D. Rehberger. The Enslaved Dataset: A Real-World Complex Ontology Alignment Benchmark Using Wikibase. In *Proceedings of the 29th ACM International Conference on Information and Knowledge Management (CIKM), October 19-23, 2020, Virtual Event, Ireland*, pages 3197–3204. ACM, 2020. 42

[65] L. Zhou, É. Thiéblin, M. Cheatham, D. Faria, C. Pesquita, C. Trojahn, and O. Zamazal. Towards evaluating complex ontology alignments. *The Knowledge Engineering Review*, 35:e21, 2020. XIII, 13, 14, 16, 17, 18, 34, 52, 57

Appendix A

Manual evaluation

Table A.1 provides the summary of manual evaluation results in terms of scores obtained for mappings of each pattern.

Table 7.2 represents the full manual evaluation of the *cmt-conference* alignment produced by PG-ARM.

Table A.1: Summary of manual evaluation results

	Pattern	N	1	0.75	0.5	0.25	0	Weigted Precision
Class	Simple	10	6	2	0	1	1	0.78
	Object occurrence restriction	22	13	1	4	2	2	0.74
	Object range restriction	6	3	0	1	0	2	0.58
	Data value restriction	0	0	0	0	0	0	-
	Data range restriction	1	1	0	0	0	0	1.00
Object Property	Simple Obj.- Obj.	3	0	3	0	0	0	0.75
	Simple Obj.-Data	0	0	0	0	0	0	-
	Inverse Obj. Prop.	0	0	0	0	0	0	-
	Domain Restriction	8	1	7	0	0	0	0.78
	Range Restriction	9	2	7	0	0	0	0.81
Data Property	Simple Data-Data	0	0	0	0	0	0	-
	Simple Data-Obj.	0	0	0	0	0	0	-
	Data Domain Restriction	0	0	0	0	0	0	-
	TOTAL	59	26	20	5	3	5	0.75

Table 7.2: Full manual evaluation of the cmt-conference alignment produced by PG-ARM

Entity1	Entity2	Relation	Evaluation
http://cmt#assignedByReviewer	http://conference#invited_by	=	0.75
http://cmt#assignExternalReviewer	http://conference#invites_co-reviewers	=	0.75
onAttribute: Relation(http://cmt#hasDecision); class: http://cmt#Acceptance	http://conference#Accepted_contribution	=	1
onAttribute: Relation(http://cmt#hasDecision); class: http://cmt#Rejection	http://conference#Rejected_contribution	=	1
http://cmt#Author	http://conference#Regular_author	=	1
http://cmt#memberOfProgramCommittee	AND{Relation(http://conference#was_a_member_of); RelationCoDomainRestriction (Class(http://conference#Program_committee); }	=	1
http://cmt#Co-author	http://conference#Contribution_co-author	=	1
http://cmt#Document	http://conference#Conference_document	=	1
http://cmt#Meta-Review	onAttribute: Relation(http://conference#has_authors); comparator: greater-than, value: 0	<	1
http://cmt#co-writePaper	AND{ RelationCoDomainRestriction (AND{ Class(http://conference#Conference_contribution); } Relation(http://conference#contributes); RelationDomainRestriction (AND{ Class(http://conference#Conference_contributor); }}	=	0.75
http://cmt#hasAuthor	AND{ RelationDomainRestriction (AND{ Class(http://conference#Conference_contribution); } RelationCoDomainRestriction (AND{ Class(http://conference#Conference_contributor); } Relation(http://conference#has_authors); }	=	0.75
http://cmt#hasProgramCommitteeMember	http://conference#has_members	=	0.75
http://cmt#hasCo-author	AND{ RelationDomainRestriction (AND{ Class(http://conference#Conference_contribution); } RelationCoDomainRestriction (AND{ Class(http://conference#Conference_contributor); } Relation(http://conference#has_authors); }	=	0.75
http://cmt#markConflictOfInterest	AND{ RelationCoDomainRestriction (AND{ Class(http://conference#Conference_contribution); } Relation(http://conference#contributes); RelationDomainRestriction (AND{ Class(http://conference#Conference_contributor); } }	=	0.75
http://cmt#Meta-Review	onAttribute: Relation(http://conference#reviews); comparator: greater-than, value: 0	<	1
onAttribute: Relation(http://cmt#acceptedBy); comparator: greater-than, value: 0	http://conference#Accepted_contribution	=	1
http://cmt#Person	http://conference#Person	=	1
http://cmt#Review	http://conference#Review	=	1
http://cmt#SubjectArea	http://conference#Topic	=	1
http://cmt#ProgramCommitteeMember	onAttribute: Relation(http://conference#was_a_member_of); class: http://conference#Program_committee	=	1
http://cmt#PaperFullVersion	http://conference#Submitted_contribution	=	0
http://cmt#Reviewer	onAttribute: Relation(http://conference#was_a_member_of); class: http://conference#Program_committee	=	0
http://cmt#User	onAttribute: Relation(http://conference#contributes); class: http://conference#Conference_contribution	=	0

Entity1	Entity2	Relation	Evaluation
http://cmt#ConferenceMember	http://conference#Conference_contributor	=	0.25
http://cmt#submitPaper	AND{ RelationCoDomainRestriction (AND{ Class(http://conference#Conference_contribution); } Relation(http://conference#contributes); RelationDomainRestriction (AND{ Class(http://conference#Conference_contributor); } }	=	0.75
http://cmt#AuthorNotReviewer	onAttribute: Relation(http://conference#contributes); class: http://conference#Submitted_contribution	=	0.5
http://cmt#writePaper	AND{ RelationCoDomainRestriction (AND{ Class(http://conference#Conference_contribution); } Relation(http://conference#contributes); RelationDomainRestriction (AND{ Class(http://conference#Conference_contributor); }}	=	0.75
http://cmt#writeReview	AND{ Relation(http://conference#contributes); RelationDomainRestriction (AND{ Class(http://conference#Reviewer); Class(http://conference#Committee_member); } RelationCoDomainRestriction (Class(http://conference#Review); }}	=	0.75
http://cmt#writtenBy	AND{{ RelationCoDomainRestriction (AND{ Class(http://conference#Reviewer); Class(http://conference#Committee_member); } Relation(http://conference#has_authors); RelationDomainRestriction (Class(http://conference#Review); }}	=	0.75
onAttribute: Property(http://cmt#paperID); , datatype: http://www.w3.org/2001/XMLSchema#integer	http://conference#Written_contribution	=	1
onAttribute: Relation(http://cmt#assignedByAdministrator); comparator: greater-than, value: 0	http://conference#Reviewer	<	1
onAttribute: Relation(http://cmt#assignedByReviewer); comparator: greater-than, value: 0	http://conference#Reviewer	<	1
onAttribute: Relation(http://cmt#assignExternalReviewer); comparator: greater-than, value: 0	http://conference#Reviewer	<	1
onAttribute: Relation(http://cmt#hasBeenAssigned); comparator: greater-than, value: 0	http://conference#Reviewer	<	1
onAttribute: Relation(http://cmt#hasDecision); comparator: greater-than, value: 0	http://conference#Reviewed_contribution	=	1
onAttribute: Relation(http://cmt#readPaper); comparator: greater-than, value: 0	http://conference#Reviewer	<	1
onAttribute: Relation(http://cmt#rejectedBy); comparator: greater-than, value: 0	http://conference#Rejected_contribution	=	1
onAttribute: Relation(http://cmt#submitPaper); comparator: greater-than, value: 0	http://conference#Contribution_1th-author	=	1
onAttribute: Relation(http://cmt#writePaper); comparator: greater-than, value: 0	http://conference#Contribution_1th-author	=	1

Entity1	Entity2	Relation	Evaluation
http://cmt#Paper	http://conference#Conference_contribution	=	0.75
http://cmt#PaperAbstract	http://conference#Extended_abstract	=	0.75
onAttribute: Relation(http://cmt#writeReview); comparator: greater-than, value: 0	http://conference#Reviewer	<	1
onAttribute: Relation(http://cmt#addedBy); comparator: greater-than, value: 0	http://conference#Reviewer	<	0
onAttribute: Relation(http://cmt#memberOfProgramCommittee); comparator: greater-than, value: 0	http://conference#Reviewer	<	0
onAttribute: Relation(http://cmt#assignedTo); comparator: greater-than, value: 0	http://conference#Reviewed_contribution	=	0.25
onAttribute: Relation(http://cmt#hasAuthor); comparator: greater-than, value: 0	http://conference#Reviewed_contribution	=	0.25
onAttribute: Relation(http://cmt#hasSubjectArea); comparator: greater-than, value: 0	http://conference#Written_contribution	=	0.5
onAttribute: Relation(http://cmt#memberOfConference); comparator: greater-than, value: 0	http://conference#Organizer	>	0.5
onAttribute: Relation(http://cmt#readByMeta-Reviewer); comparator: greater-than, value: 0	http://conference#Reviewed_contribution	=	0.5
onAttribute: Relation(http://cmt#readByReviewer); comparator: greater-than, value: 0	http://conference#Reviewed_contribution	=	0.5
http://cmt#ExternalReviewer	onAttribute: Relation(http://conference#invited_by); comparator: greater-than, value: 0	=	0.75