Minimal Coverage for Ontology Signatures

David Geleta, Terry R. Payne, and Valentina Tamma

Department of Computer Science, University of Liverpool, UK
{d.geleta|t.r.payne|v.tamma}@liverpool.ac.uk

Abstract. An ontology *signature* (set of entities) can express more than its constituent concept, role and individual names, since *rewriting* permits defined entities to be replaced by syntactically different, albeit semantically equivalent definitions. Identifying whether a given signature permits the definition of a particular entity is a well-understood problem, while determining the smallest (minimal) signature that covers a set of entities (i.e. a task signature) poses a challenge: the complete set of alternative definitions, or even just their signature, needs to be obtained, and all combinations of such definition signatures need to be explored, for each of the entities under consideration. In this paper, we present and empirically evaluate our novel approach for efficiently computing an approximation of *minimal signature cover sets.*

1 Introduction

An ontology provides a reference vocabulary for a given domain of interest, where the meaning of the terms in the vocabulary (entities) is *defined* inductively in terms of other entities [1]. Beth definability [2, 6] is a well-known property from classical logic that has also been studied in the context of Description Logics (DLs); it relates the notion of *implicit definability* to the one of *explicit definability*, by stating that every implicitly defined concept is also explicitly definable [11]. In definitorially complete DLs ontologies (i.e. those modelled in a dialect where the *Beth definability* property holds), defined ontological entities can be rewritten into syntactically different, albeit semantically equivalent forms, thus it is possible to convey the meaning of an entity without using its actual name [11].

For example, in the ontology $\mathcal{O} = \{\mathsf{C} \equiv \mathsf{A} \sqcup \mathsf{B}, \mathsf{C} \equiv \mathsf{D}, \mathsf{A} \sqsubseteq \neg\mathsf{B}, \mathsf{E} \sqsubseteq \exists\mathsf{r}.\top\}$, the concept C is defined explicitly, i.e. $\mathsf{C} \equiv \mathsf{A} \sqcup \mathsf{B}$. Additionally, the concept A is implicitly defined in $\mathcal{O}$ by the set of general concept inclusions $\{\mathsf{C} \equiv \mathsf{A} \sqcup \mathsf{B}, \mathsf{A} \sqsubseteq \neg\mathsf{B}\}$. Thus, A can be explicitly defined by the axiom $\mathsf{A} \equiv \mathsf{C} \sqcap \neg\mathsf{B}$, where $\Sigma = \{\mathsf{C}, \mathsf{B}\}$ is a *definition signature* (DS) of A.

A concept or role can either be defined explicitly or implicitly in an ontology, or it could be *undefined* (and hence not rewritable, e.g. E, r). For instance, apart from the concept E, Σ *provides coverage* for all concept names of $\mathcal{O}$, as in addition to the asserted entities, both A and D are definable by an axiom whose signature is in Σ. Therefore, a signature potentially enables the expression of not only its asserted concept, role and individual names, but also of those defined entities whose definition is permitted with the given signature.

Semantic interoperability between individually designed ontologies is typically hindered by heterogeneity, as distinct ontologies typically differ in their vocabularies and in the meaning they associate with particular entities [3]. *Ontology matching (alignment)* resolves heterogeneity between different ontologies by producing an alignment, i.e. a set of correspondences that describe relationships between semantically related entities of distinct ontologies. *Ontology alignment negotiation* has become an established and active research area that is concerned with supporting opportunistic communication within open environments [9, 10]. In order to be able to carry out meaningful communication, different systems must cooperatively establish a mutually acceptable alignment, whilst adhering to internal preferences, without compromising confidential knowledge, and prevent alignment-based conservativity violations to occur [8]. The mutual alignment emerges as a result of a bilateral negotiation between two systems, whereby it is clearly beneficial to *minimise* the considered correspondences (i.e. the aligned part of an ontology signature), in order to reduce the overall cost of the process, and to support privacy and conservativity constraints of the interacting parties.

van Harmelen and colleagues [12] have shown that semantic interoperability tasks can be characterised (amongst other parameters) by their signature, thus the prerequisite for performing a knowledge-based task is that the tasks's signature must be *covered* by the terms available to the party that performs the given task. In order to determine whether a given *task signature* (e.g. a query signature) is covered by a given *ontology signature*, each entity of the task signature must be individually examined; an entity is covered either if it appears in the ontology signature, or if it is rewritable using only the members of the ontology signature. Although task coverage is trivial to establish, determining the minimal signature that covers a given task signature poses a challenge, as the complete set of rewriting forms (definitions) needs to be known, and all combinations of such definition signatures (the sets of entities that permit the rewriting of a defined entity) are required to be explored, for each entity in question.

In our previous work, we have presented a pragmatic approach to computing the complete set of Definition Signatures for a given ontology [4]. In this paper we introduce the signature coverage problem and a novel algorithm that can efficiently compute an approximation of the smallest set of entities that covers a given task signature, starting from a previously obtained complete set of DSs. The paper casts the signature coverage problem as a set coverage problem, which is in Section 2, whilst Section 3 introduces the signature coverage problem. Section 4 presents our approach to approximate minimal cover sets. The approach is empirically evaluated in Section 5, that describes the experimental framework and the results. Finally, Section 6 outlines future work and concludes the paper.

2 Background

In this section we review two classical problems that address the issue of *coverage*: the set coverage problem and the minimal functional dependency cover. The *set coverage problem* (or minimal set cover problem) is a classic problem in combinatorics, complexity theory and computer science, in general [14]. Let $\mathcal{U}$ be a set of elements (referred to as the *universe*) and $\mathcal{S}$ a collection of subsets of $\mathcal{U}$, whose union equals the universe. The set cover problem identifies the smallest, *minimal* sub-collection $\mathcal{C} \subseteq \mathcal{S}$, called the *cover set*, such that the union of sets in $\mathcal{C}$ covers $\mathcal{U}$ ($\forall x \{x \in \mathcal{U} | x \in \mathcal{C}\}$).

Example 1 (Minimal set cover problem). Consider the set $\mathcal{U} = \{1, 2, 3, 4, 5\}$, and let $\mathcal{S}$ be the collection $\{\mathcal{S}_1, \mathcal{S}_2, \mathcal{S}_3, \mathcal{S}_4\}$, where $S_1 = \{1, 2, 3\}$, $\mathcal{S}_2 = \{1, 2\}$, $\mathcal{S}_3 = \{3, 4\}$ and $\mathcal{S}_4 = \{4, 5\}$. The union of subsets of $\mathcal{S}$ contains all members of $\mathcal{U}$, thus $\mathcal{U}$ can be covered by an $S' \subseteq \mathcal{S}$. Although there are a number of possible solutions, there is only one minimal cover set, $\{S_1, S_4\}$.

Finding the minimal cover set is an NP-complete problem, however, there is a greedy algorithm that is able to find *approximations* (i.e. not necessarily minimal, but small cover sets) in polynomial time [14]. In the *weighted set cover* problem, each set $\mathcal{S}_i \in \mathcal{S}$ is assigned a *weight* $w(\mathcal{S}) \geq 0$, and in this case, the goal is to find a cover set $\mathcal{C}$ with the minimal total weight $\sum_{S \in \mathcal{C}} w(S)$ (where the weight of a set does not correspond to its cardinality but emerges from the particular context where the set is used).

Another classical problem that has influenced the approach presented in this paper comes from relational database theory, and concerns finding the minimal functional dependency cover. A *functional dependency (FD)* is a constraint between two sets of attributes in a relation from a database [13]. For instance, given two attribute sets X and Y, then the FD $X \rightarrow Y$ means that the values of the attribute set Y are determined by the values of X, or in other words, two tuples in a database sharing the same values of X would also share the same values for Y. The *closure* of a set of attributes X with respect to a set of FDs $\mathcal{F}$ is the set X^+ of all attributes that are functionally determined by X using the closure of $\mathcal{F}$, denoted by $\mathcal{F}^+$. Before computing the closure, a set of FDs $\mathcal{F}$ is usually *normalised* by exhaustively applying inference rules (e.g. reflexivity, transitivity and augmentation) [13], as illustrated by the following example:

Example 2 (FD set attribute closures). Let us consider a set of FDs $\mathcal{F}$ such that

$$\mathcal{F} = \{(1)\ A \rightarrow B,\ (2)\ C \rightarrow E,\ (3)\ E \rightarrow F,\ (4)\ A, C \rightarrow D\}$$

$\mathcal{F}$ is already normalised (in third normal form), i.e. each FD contains exactly one attribute on the right-hand side (RHS), and each FDs' left-hand side (LHS) is irreducible. The closure of all attributes in $\mathcal{F}$ is given as follows:

1. $A^+ : A, B$ (A by reflexivity, B by (1))
2. $B^+ : B$ (B by reflexivity)
3. $C^+ : C, E, F$ (C by reflexivity, E by (2), F by transitivity and (2, 3))
4. $D^+ : D$ (D by reflexivity)
5. $F^+ : F$ (F by reflexivity)
6. $(A, C)^+ : A, B, C, D, E, F$ (A, C by reflexivity, B by (1), D by (4), E by (2), F by transitivity and (2, 3))

The closure of a set of attributes with respect to a set of FDs allows us to determine the minimal cover for a set of functional dependencies. A set of functional dependencies $\mathcal{F}$ covers another set of FDs $\mathcal{G}$ if every functional dependency in $\mathcal{G}$ can be inferred from $\mathcal{F}$, i.e. if $\mathcal{G}^+ \subseteq \mathcal{F}^+$. $\mathcal{F}$ is a *minimal cover* of $\mathcal{G}$ if $\mathcal{F}$ is the smallest set of functional dependencies that covers $\mathcal{G}$. It can be proven that every set of functional dependencies has a minimal cover, however this is not unique, as there may be more than one minimal cover.

3 The Signature Coverage Problem

The signature coverage problem determines whether a given *task signature* ($\mathcal{S}$) is covered by a *restricted signature* ($\mathcal{R}$), where both are subsets of the same vocabulary, i.e. the ontology signature ($\mathcal{R}, \mathcal{S} \subseteq \mathsf{Sig}(\mathcal{O})$). The restricted signature $\mathcal{R}$ could be obtained by considering only those entities in $\mathcal{O}$) that are mapped in an alignment. A task signature is said to be *covered* if all of its constituent entities are covered; by default, i.e. considering *explicit coverage*, this requires that $\mathcal{S} \subseteq \mathcal{R}$. Beth definability supports *implicit coverage*, where a task signature entity is replaceable by a definition axiom, if, given its signature Σ, $\Sigma \subseteq \mathcal{R}$. In order to achieve implicit coverage, the complete set of DSs needs to be obtained, where the number of possible rewritings (thus the number of unique DSs) of a defined concept or role is potentially exponential in the size of the ontology. Definition signatures may contain redundant elements, and could be as large as the ontology signature, thus we introduce the notion of signature minimality, that aims at minimising the size of a signature, by eliminating superfluous entities.

Definition 1 (Minimal Definition Signature (MDS)). *A signature Σ is a minimal definition signature of a defined entity e under a TBox $\mathcal{T}$, if none of its proper subsets are definition signatures of e.*

In previous work we have presented a pragmatic approach to compute in practice all Minimal Definition Signatures (MDSs). The approach focusses on the feasible defined entities (where given pre-defined complexity bounds all MDSs are computable) described using a DL language for which the Beth definability property holds [4]. Individual names can only be covered *explicitly* by an asserted entity, however, definable signature entities (concepts and roles) can also be covered *implicitly* as follows:

Definition 2 (Explicitly or implicitly covered entity). *Given an ontology $\mathcal{O}$, a task signature $\mathcal{S}$, and a restricted signature $\mathcal{R}$ such that $\mathcal{S}, \mathcal{R} \subseteq \mathsf{Sig}(\mathcal{O})$, an entity $e \in \mathcal{S}$ is covered explicitly by $\mathcal{R}$ iff $e \in \mathcal{R}$; or covered implicitly by $\mathcal{R}$, if there exists a DS Σ^e, such that $\Sigma \subseteq \mathcal{R}$; otherwise e is uncovered.*

A defined concept or role can simultaneously be covered explicitly and implicitly, thus a task signature entity $e \in \mathcal{S}$ may assume one of the four different *coverage status* w.r.t. a restricted signature $\mathcal{R}$, as shown in Table 1. Determining whether a given task signature is coverable by a particular, restricted signature is the trivial process of identifying the coverage status of each task signature entity;

Coverability			Required Coverage
uncoverable		$e \notin \mathcal{R} \wedge \Sigma \not\subseteq \mathcal{R}$	explicit coverage: $e \in_? \mathcal{R}$
coverable	explicitly only	$e \in \mathcal{R} \wedge \Sigma \not\subseteq \mathcal{R}$	
	explicitly and implicitly	$e \in \mathcal{R} \wedge \Sigma \subseteq \mathcal{R}$	explicit and implicit coverage: $e \in_? \mathcal{R}^+$
	implicitly only	$e \notin \mathcal{R} \wedge \Sigma \subseteq \mathcal{R}$	

Table 1: Entity coverage status

i.e. for each entity $e \in \mathcal{S}$, we search for an MDS Σ^e such that $\Sigma^e \subseteq \mathcal{R}$. The set of entities covering all members of a task signature $\mathcal{S}$ is defined as follows:

Definition 3 (Cover set). *Given an ontology $\mathcal{O}$, a task signature $\mathcal{S}$, and a restricted signature $\mathcal{R}$ such that $\mathcal{S}, \mathcal{R} \subseteq \mathsf{Sig}(\mathcal{O})$, $\mathcal{C}$ is a cover set of $\mathcal{S}$ with respect to $\mathcal{R}$, if and only if $\mathcal{C} \subseteq \mathcal{R}$ and $\forall e\{e \in \mathcal{S} | e \in \mathcal{C} \vee \exists \Sigma^e | \Sigma^e \subseteq \mathcal{C}\}$.*

In other words, a cover set is a DS of all entities of the task signature. As an ontology signature can cover more than its constituent entities, due to the fact that a given signature may permits some defined entities to be implicitly covered, we adopt the notion of *closure* from FD computation, to provide a representation which describes *the set of all entities covered* by a given signature:

Definition 4 (Signature closure). *Given an ontology $\mathcal{O}$, and a signature $\mathcal{X}$ such that $\mathcal{X} \subseteq \mathsf{Sig}(\mathcal{O})$, the signature closure $\mathcal{X}^+$ contains all explicitly and implicitly covered entities of $\mathsf{Sig}(\mathcal{O})$ by $\mathcal{X}$, i.e. $\forall e\{e \in \mathsf{Sig}(\mathcal{O}) | e \in \mathcal{X} \vee \exists \Sigma^e | \Sigma^e \subseteq \mathcal{X}\}$.*

This permits a more succinct definition of coverage: a task signature $\mathcal{S}$ is covered by a set $\mathcal{C}$ iff $\mathcal{S} \subseteq \mathcal{C}^+$. Once it has been established, that the $\mathcal{S}$ is coverable by $\mathcal{R}$, the problem is to identify the *smallest subset* $\mathcal{C} \subseteq \mathcal{R}$ which covers $\mathcal{S}$. This is referred to as the minimal cover set and defined as follows:

Definition 5 (Minimal cover set). *Given an ontology $\mathcal{O}$, a task signature $\mathcal{S}$, a restricted signature $\mathcal{R}$ such that $\mathcal{S}, \mathcal{R} \subseteq \mathsf{Sig}(\mathcal{O})$, and the set $\mathcal{C}$ which covers $\mathcal{S}$ with respect to $\mathcal{R}$, $\mathcal{C}$ is minimal if and only if there is no other cover set $\mathcal{C}' \subseteq \mathcal{R}$ such that $|\mathcal{C}'| < |\mathcal{C}|$.*

There can be more than one, unique minimal cover set, i.e. two sets with the same cardinality whose complement is not an empty set. Finding a minimal cover set is an non-polynominal problem, because it requires all cover sets to be identified, by exhaustively testing each subset of the power set of the ontology signature, in order to find all valid covers and select the one with the minimum cardinality. *Approximation algorithms* are commonly used for problems with NP time complexity, such as the set cover problem, to provide sub-optimal solutions in polynomial-time. The *greedy algorithm design* is one of the standard techniques for approximation algorithms [14]. The next example illustrates the signature cover problem, and outlines a cover set approximation approach.

Example 3 (Signature Coverage). Let $\mathcal{O}$ be an ontology, $\mathcal{S}$ a task signature, $\mathcal{R}$ a restricted signature, and M the complete set of MDSs of each defined entity of $\mathcal{S}$ (an MDS $m \in M$ is represented as the tuple $\langle e, \Sigma \rangle$, where e is the defined concept or role name,and Σ denotes the MDS), where $\mathcal{S}, \mathcal{R} \subseteq \mathsf{Sig}(\mathcal{O})$, such that

– $\mathsf{Sig}(\mathcal{O}) = \{\mathsf{A}, \mathsf{B}, \mathsf{C}, \mathsf{D}, \mathsf{E}, \mathsf{F}, \mathsf{r}, \mathsf{s}, \mathsf{q}\}$
– $M = \{\langle \mathsf{C}, \{\mathsf{A}, \mathsf{B}\}\rangle, \langle \mathsf{C}, \{\mathsf{E}, \mathsf{r}\}\rangle, \langle \mathsf{C}, \{\mathsf{q}\}\rangle, \langle \mathsf{B}, \{\mathsf{D}\}\rangle, \langle \mathsf{D}, \{\mathsf{B}\}\rangle, \langle \mathsf{s}, \{\mathsf{r}\}\rangle\}$
– $\mathcal{S} = \{\mathsf{B}, \mathsf{C}, \mathsf{D}, \mathsf{E}, \mathsf{s}, \mathsf{q}\}$
– $\mathcal{R} = \{\mathsf{A}, \mathsf{B}, \mathsf{C}, \mathsf{D}, \mathsf{E}, \mathsf{r}, \mathsf{q}\}$

Without accounting for definability, i.e. by only considering explicit coverage, the $\mathcal{R}$ does not cover the task signature because $\mathcal{S} \setminus \mathcal{R} \neq \emptyset$. However, considering implicit coverage shows that the closure of the restricted signature is $\mathcal{R}^+ = \{\mathsf{A}, \mathsf{B}, \mathsf{C}, \mathsf{D}, \mathsf{E}, \mathsf{r}, \mathsf{s}, \mathsf{q}\}$, thus $\mathcal{S}$ can be covered by $\mathcal{R}$, because $\mathcal{S} \subseteq \mathcal{R}^+$. Following a naive, greedy approach, one may select those entities that appear both in $\mathcal{S}$ and $\mathcal{R}$ as these entities can be covered explicitly, i.e. $C_1 = \mathcal{S} \cap \mathcal{R} = \{\mathsf{C}, \mathsf{D}, \mathsf{E}, \mathsf{q}\}$; then attempt to cover the remaining task signature entities, by adding a corresponding MDSs for each uncovered task signature entity; in this case covering s by adding r to C_1, as there is an MDS $\langle \mathsf{s}, \{\mathsf{r}\}\rangle$ thus s is implicitly definable by the signature $\{\mathsf{r}\}$. As a result $C_1 = \{\mathsf{B}, \mathsf{C}, \mathsf{D}, \mathsf{E}, \mathsf{r}, \mathsf{q}\}$ covers $\mathcal{S}$. However, the smallest cover set is $C_2 = \{\mathsf{B}, \mathsf{E}, \mathsf{r}, \mathsf{q}\}$, because $C_2^+ = \{\mathsf{B}, \mathsf{C}, \mathsf{D}, \mathsf{E}, \mathsf{r}, \mathsf{s}, \mathsf{q}\}$, $\mathcal{S} \subseteq C_2^+$, and $|C_1| > |C_2|$.

In example 3, a naive, greedy approach (*Greedy #1*) has produced a non-minimal cover set C_1, which was an approximation of the minimal cover C_2. The cover set C_1 can be improved by removing redundant entities (resulting in the set $C_1' = C_2$), i.e. producing a non-redundant cover set:

Definition 6 (Non-redundant cover set). *Given an ontology $\mathcal{O}$, a task signature $\mathcal{S}$, a restricted signature $\mathcal{R}$ such that $\mathcal{S}, \mathcal{R} \subseteq \mathsf{Sig}(\mathcal{O})$, and $\mathcal{C}$ which covers $\mathcal{S}$ w.r.t. $\mathcal{R}$, $\mathcal{C}$ is non-redundant iff none of its proper subsets cover $\mathcal{S}$.*

Non-redundant cover sets are typically small, however, as there can be more than one non-redundant cover set with different cardinality, a non-redundant cover set is not necessarily the minimal cover set. It is worth noting that every minimal cover set is also a non-redundant set.

4 Approximating Minimal Cover Sets

In order to tackle the exponential time complexity of the minimal signature cover problem, we introduce a greedy, approximation algorithm that provides a sub-optimal solution in polynomial-time; where the resulting cover set is always non-redundant. The basic idea behind the approach is that by starting from an empty set, the cover set is built up incrementally until all task signature members are covered, however, instead of selecting individual entities from the restricted signature, at each iteration the approach selects an entity set. The entity sets that are being considered are MDSs, because individual entities typically only provide explicit coverage (for themselves or their synonyms), while MDSs can cover defined entities implicitly (in addition to explicitly covering all those task signature entities that appear in the MDS as well). The selection is made by assigning a *cost* and *value* score to each MDS, and then picking the MDS which provides the maximum value and the minimum cost with respect to the task signature and the incomplete cover set, prioritising on the value score. The cost quantifies the number of entities required to be added to the cover set (i.e. the set difference of the cover set and the particular MDS), while the value represents the number of entities that the given signature covers (an MDS can be a DS for more than one defined entity, thus it can cover several task signature entities). In case there are more than one MDSs with the same cost and value, a random MDS is selected.

In order to evaluate the *actual value* of a given MDS, i.e. the set of all entities of the task signature that the MDS covers either explicitly or implicitly, similarly to FDs, its *closure* needs to be identified, thus we represent MDSs in the form of FDs to facilitate this notion. There is a strong resemblance between the concept of an FD and an MDS, meaning that an MDS can be thought of as a dependency between entities of an ontology, where the relation between the signature of the LHS and the entity on the RHS is implicit definability. For example, the MDS $\Sigma^{\mathsf{C}} = \{\mathsf{A}, \mathsf{B}\}$ which defines concept C using entities $\{\mathsf{A}, \mathsf{B}\}$ may be represented as $m : (\mathsf{A}, \mathsf{B} \rightarrow \mathsf{C})$; such an MDS is referred to as an fMDS, and defined as follows:

Definition 7 (fMDS). *Given a defined entity e, and its minimal definition signature Σ, where $e \in \mathsf{Sig}(\mathcal{O})$ and $\Sigma \subseteq \mathsf{Sig}(\mathcal{O})$, the corresponding fMDS is the function $m : (\Sigma \rightarrow e)$, which, given the entity set Σ, implicitly covers e.*

Analogously to functional dependencies, the closure of an fMDS is computed from the *set of all fMDSs*, by identifying all relevant MDSs:

Definition 8 (fMDS closure). *Given an fMDS $m_i : (\Sigma \rightarrow e)$, and a set of fMDSs M where $m_i \in M$, the closure of m_i is the function $m_i^+ : (\Sigma \rightarrow E)$ such that*

$$E = \Sigma \cup \{e\} \cup (\bigcup \forall m_j^{+RHS} \{m_j \in M | m_j^{+LHS} \subseteq m_i^{+LHS}\})$$

where m^{+LHS} denotes the signature Σ, and m^{+RHS} refers to signature E.

The closure of a set of fMDSs M is the set M^+, where each $m_i^+ \in M^+$ is the closure of the corresponding $m_i \in M$; this is illustrated by the next example:

Example 4 (fMDS set closure). Let M be a set of fMDS such that

- $M = \{m_1 : (\mathsf{A}, \mathsf{B} \rightarrow \mathsf{C}), m_2 : (\mathsf{B} \rightarrow \mathsf{D})\}$
- $M^+ = \{m_1^+ : (\mathsf{A}, \mathsf{B} \rightarrow \mathsf{A}, \mathsf{B}, \mathsf{C}, \mathsf{D}), m_2^+ : (\mathsf{B} \rightarrow \mathsf{B}, \mathsf{D})\}$

The closure of fMDSs (M^+) is computed as follows:

1. m_1^{+LHS} in addition to implicitly covering concept C, also explicitly covers concepts A, B, hence $m_1^{+RHS} = m_1^{RHS} \cup \{\mathsf{A}, \mathsf{B}\}$;
2. m_1^+ implicitly covers D as $m_2^{+LHS} \subseteq m_1^{+RHS}$ thus $\mathsf{D} \in m_1^{+LHS}$;
3. m_2 explicitly covers concepts B, hence $m_2^{+RHS} = m_2^{RHS} \cup \{\mathsf{B}\}$;
4. no more MDSs apply, thus the closure is complete.

The cost and value calculation of an fMDS is formalised as follows:

Definition 9 (fMDS value and cost). *Given an fMDS m, an ontology $\mathcal{O}$, a task signature $\mathcal{S}$, a cover set $\mathcal{C}$, and M the complete set of fMDSs in $\mathcal{O}$, where $m, \mathcal{S} \subseteq \mathcal{O}$, the value and cost of m with respect to $\mathcal{S}$ and $\mathcal{C}$ is given by the value function $v(m) = |\mathcal{R} \setminus \mathcal{C}^+ \cap m^{+RHS}|$, and the cost function $c(m) = |\mathcal{C}^+ \setminus m^{LHS}|$, where both $v(m)$ and $c(m)$ assign a natural number $i \in \mathbb{N}_0$ to m.*

Algorithm 1: ComputeMinimalSignatureCover($\mathcal{O}, \mathcal{R}, \mathcal{S}, M$)

```
Input  : $\mathcal{O}$: ontology; $\mathcal{S}$: task signature; $\mathcal{R}$: restricted signature;
         $M$: the complete set of MDSs of each defined entity $e \in \mathcal{O}$
Output: $\mathcal{C}$: non-redundant cover set of $\mathcal{S}$ w.r.t. $\mathcal{R}$ if and only if $\mathcal{S} \subseteq \mathcal{R}^+$
1  $\mathcal{C} \leftarrow \mathcal{C} \cup \forall e\{e \in \mathcal{S} \cap \mathcal{R} | e \notin m_i^{RHS} | m_i \in M\}$
2  $M^+ \leftarrow \mathsf{Initialise}(M)$
3  $\mathcal{C}^+ \leftarrow \mathsf{ComputeSignatureClosure}(\mathcal{C}, M^+)$
4  $\overline{M^+} \leftarrow M^+ \setminus \forall m_i\{m_i \in M^+ | m_i^{LHS} \subseteq \mathcal{C}^+\}$
5  while $(\mathcal{S} \setminus \mathcal{C}^+) \neq \emptyset$ do
6      $\mathcal{V} \leftarrow \mathsf{ComputeValueCostVector}(M^+, \mathcal{S}, \mathcal{C}^+)$
7      $m' \leftarrow$ select an $m \in \overline{M^+}$ according to $\mathcal{V}$, with max $v(m)$, and min $c(m)$
8      $\mathcal{C} \leftarrow \mathcal{C} \cup m'^{LHS}$
9      $\mathcal{C}^+ \leftarrow \mathsf{ComputeSignatureClosure}(\mathcal{C}, M)$
10     $\overline{M^+} \leftarrow \overline{M^+} \setminus (\{m'\} \cup \forall m_i\{m_i \in \overline{M^+} | m_i^{LHS} \subseteq \mathcal{C}^+\})$
11 end
12 return $\mathcal{C}$
```

Algorithm 1 formalises the approximation approach, which employs two subroutines for computing the *closure of signatures*, and the *closure of fMDSs*. The algorithm first applies an *optimisation heuristic*, which reduces the search space by initialising cover $\mathcal{C}$ with the only explicitly coverable entities of $\mathcal{S}$ (line 1). Next M, which is used as the search space, is initialised with the complete set of fMDSs. In addition, the process generates an 'artificial' MDS and stores it in M^+, for each entity that can be covered both explicitly and implicitly. For instance, given a concept A, the generated fMDSs is $\mathsf{m} : (\mathsf{A} \rightarrow \mathsf{A})$, i.e. the entity can cover itself. By including such artificial fMDSs that do not originate from actual MDSs, the algorithm ensures that the search space is complete, i.e. for each $e \in \mathcal{S}$ the search space M^+ includes *all* possible ways of cover. The initialisation is concluded by computing the fMDS set closure (line 2). Before the process begins the search, $\mathcal{C}^+$, the cover closure is computed. This facilitates the *termination condition of the search* process (line 5) that halts the algorithm when $\mathcal{S}$ is covered (i.e. $\mathcal{S} \setminus \mathcal{C}^+ = \emptyset$). Then $\overline{M^+}$ is created as a copy of M^+, the former is the actual search space which is continuously pruned at each iteration (to optimise the process, by reducing the search space and subsequently the effort required to calculate the cost and value scores of fMDSs), while the latter is left intact for the purpose of computing signature closures during the search. $\overline{M^+}$ is pruned out by removing any fMDS whose value (and cost) w.r.t. $\mathcal{C}$ is zero (line 4). During the search (line 5-11), the value and cost of each fMDS in $m_i \in \overline{M^+}$ is evaluated w.r.t. the cover (line 6), then the best fMDS is selected (line 7) and the LHS of the fMDS (i.e. the MDS) is added to the cover (line 8). The cover is then reevaluated by updating its closure (line 9), finally $\overline{M^+}$ is pruned according to the updated cover set. These steps are repeated until $\mathcal{S}$ is covered, at which point the algorithm returns the completed cover.

The algorithm always finds a non-redundant cover set, this is ensured by the selection function, and the fact that the entities added to the cover are MDSs, i.e. already minimal entity sets that are required to cover an other entity. At the worst case, the process covers at least one entity at each iteration, thus the maximum number of steps performed by the algorithm is n, where $n = |\mathcal{S}|$. As both subroutines employed by this algorithm have polynomial time computational complexity, it holds that the overall complexity is polynomial in the size of the input as well. Moreover, as both of subroutines terminate, and the halting condition (line 5) suspends the main loop of Algorithm 1 when the cover set is complete, it follows that the Algorithm 1 also terminates. A more exhaustive description of the presented algorithms can be found in [5].

5 Empirical Evaluation

In this section, we empirically determine how effective our approximation approach is in finding cover sets. The evaluation tested the *hypothesis* that the presented approach (*Greedy #2*), by considering both explicit and implicit coverage, produces a cover set that, although not minimal, is still significantly smaller than cover sets obtained by only explicit coverage. Thus approximations of minimal cover sets are typically smaller than explicit covers, if the given task signature contains defined entities w.r.t. a restricted signature (clearly, for a task signature which lacks defined entities only explicit coverage is possible).

The *evaluation corpus* was assembled from several semantically rich OWL ontologies that are commonly used for empirical evaluation in the ontology matching and alignment negotiation literature. We have selected 7 small ontologies (average 114.57 concepts and roles, and 307.57 axioms per ontology) from the *Conference* dataset[1], which describes the conference organisation domain; and 2 large (average 13397.00 entities, and 14663.00 axioms) ontologies from the *Large biomedical* dataset[2]. For every concept and role in each ontology, we have pre-computed the definability status and the complete set of MDSs. Table 2 presents a summary of the corpus, showing the DL expressivity, the number of logical axioms, number of concept and roles ($\mathcal{C} \cup \mathcal{R}$) in the ontology signature, the ratio of defined entities to all entities ($Def\%$), and the average number of different MDSs per defined entity ($\mathcal{M}$). Both datasets contain ontologies with varying level of definability, as shown by the ratio of defined ontology signature entities and the number of different MDSs per entity.

The *experimental framework* was implemented in Java; the OWL API [7] was used for ontology manipulation; entity definability status, and MDSs were computed using the OntoDef API [4]. In all experiments, for each task signature, we have computed cover sets by using the approximation approach. We have only considered *coverable* task signatures (i.e. $\mathcal{S} \subseteq \mathcal{R}^{+}$), thus in all cases, the restricted signature $\mathcal{R}$ was equivalent to the T-Box signature, while the task signature $\mathcal{S}$ was allocated several differently sized T-Box signature subsets (i.e. $\mathcal{R} = \mathsf{Sig}(\mathcal{T})$, and $\mathcal{S} \subseteq \mathsf{Sig}(\mathcal{T})$). Varying the composition of only one of the two signatures simplified both the experiment conduct and the result analysis process. Experiments were conducted with 8GB maximum memory allocated for the

[1] `http://oaei.ontologymatching.org/2014/conference/index.html`
[2] `http://oaei.ontologymatching.org/2014/largebio/index.html`

Ontology	$\mathcal{DL}$ Expressivity	Axioms	$(\mathcal{C} \cup \mathcal{R})$	*Def*%	$\mathcal{M}$	*Ideal Cover* *cov*	*Greedy #2* *cov*	Time
				Conference corpus				
cmt	$\mathcal{ALCIN}(\mathcal{D})$	226	88	50.00%	1.09	50.00%	72.73%	3.62 ms
conference	$\mathcal{ALCHIF}(\mathcal{D})$	285	123	57.72%	1.54	42.28%	70.73%	11.79 ms
confOf	$\mathcal{SIN}(\mathcal{D})$	196	74	12.16%	3.33	87.84%	89.19%	0.43 ms
edas	$\mathcal{ALCOIN}(\mathcal{D})$	739	153	26.14%	2.80	73.86%	86.27%	8.70 ms
ekaw	$\mathcal{SHIN}$	233	106	28.30%	1.00	71.70%	85.85%	2.30 ms
iasted	$\mathcal{ALCIN}(\mathcal{D})$	358	181	17.68%	1.75	82.32%	87.85%	3.18 ms
sigkdd	$\mathcal{ALEI}(\mathcal{D})$	116	77	25.97%	1.55	74.03%	81.82%	1.12 ms
	AVG.	**307.57**	**114.57**	**31.14%**	**1.87**	**68.86%**	**82.06%**	**4.45 ms**
				LargeBio corpus				
NCI_fma	$\mathcal{ALC}$	9083	6551	29.98%	1.32	70.02%	70.02%	3.09 s
SNOMED_fma	$\mathcal{ALER}$	20243	13430	21.47%	1.09	78.54%	78.56%	8.64 s
	AVG.	**14663.00**	**13397.00**	**25.72%**	**1.16**	**74.28%**	**74.29%**	**5.86 s**

Table 2: Comparing ideal covers with approximations produced by approach #2, for covering the entire ontology signature.

JVM, running on a machine equipped with 16GB RAM and a 4-core 2.9 GHz 64-bit processor architecture.

Experiment 1: Full signature. This experiment compares the size of the approximated covers to the *ideal cover*. The only cover which is potentially minimal and can be computed efficiently, is obtainable when the task requires the entire signature of an ontology to be covered ($\mathcal{S} = \mathsf{Sig}(\mathcal{O})$). This special case provides the opportunity to evaluate the difference between an actual, and an approximated minimal cover set. The ideal cover is obtained by removing all non-redundant, defined entities from the ontology signature. Considering non-redundancy in role removal is necessary in order to avoid removing those entities that are both defined, and provide the only DS to another entity. For example, the axiom $\mathsf{r} \equiv \mathsf{s}$ implies that both roles r and s are defined, however removing both entities from the ideal cover would make them both uncoverable. Table 2 presents the experiment results, showing the size of the cover set in relation to the an explicit cover (which is always equivalent to the task signature hence $cov = \frac{|C|}{|\mathcal{S}|}$), while the RHS partition present the result obtained by the greedy algorithm, in terms of the cover to task signature size ratio, and computation time (wall time). The approach achieves considerable reduction in all ontologies, on average the approximated cover is 82.06% of the explicit cover in the Conference, and 68.86% in the LargeBio corpus (which only falls short by 0.01% from the optimal solution, i.e. the ideal cover).

Experiment 2: Varying signatures. The second experiment was carried out using different task signature sizes, in order to assess the reduction provided by a minimal cover in comparison with the baseline (explicit cover), and to evaluate the computation time on a wider scale of possible tasks sizes. The experiment included 20 test cases for each ontology, where the task to ontology signature ratio ranged between 100% and 5%. Due to the fact that cover computation includes a non-deterministic part, where a random choice is made to select an MDS from a set of equally good options (MDSs with the same value

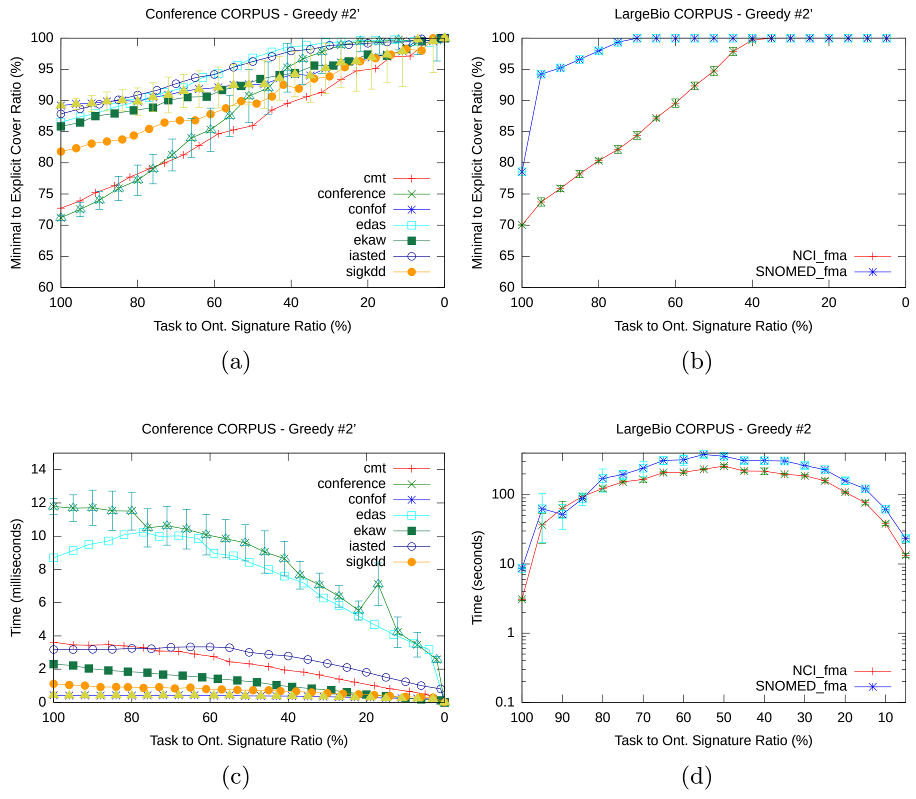

Fig. 1: Cover set *sizes* (*a, b*) and *computation times* (*c, d*) in the corpus.

and cost scores), each test case was repeated over 100 times (we have tested several different repetition counts, and by comparing their relative standard deviation established that 100 repetitions were sufficient for both datasets). Figure 1 (a, b) presents the *cover set cardinality* results, where the y-axis represents the approximated minimal cover to task signature ratio ($\frac{|\mathcal{C}|}{|\mathcal{S}|}$), and the x-axis shows the task signature to ontology signature ratio ($\frac{|\mathcal{S}|}{|\mathsf{Sig}(\mathcal{O})|}$); for brevity, error bars are only shown for the ontologies with the highest and lowest covers. The approach achieves varying level of reduction in all ontologies, where the efficiency decreases with the size of the task signature (due to the lower probability of the signature containing defined entities). Figure 1 (c, d) presents the *computation time* (wall time) results, where the y-axis shows the time (either in milliseconds for the Conference, or in seconds for the LargeBio corpus), and the x-axis shows the task to ontology signature size ratio. The results suggest that the approach is in practice feasible for both small and large ontologies.

6 Conclusions and Future Work

In this paper, we have introduced and characterised the ontology signature coverage problem, which is a non-polynomial time problem that concerns whether an ontology signature can be covered by another signature, under a given ontology. Furthermore, we have presented and empirically evaluated a novel approach that, by exploiting the notion of Beth-definability in DL ontologies and using the pre-computed, complete set of different MDSs, provides a sub-optimal solution to the minimal signature cover problem. The evaluation has confirmed that, although the resulting covers are not necessarily minimal, the presented greedy approximation approach provides significant reduction in cover set size than only explicit coverage. The approach identifies a single cover, however, there can be more than one minimal cover set, hence as future work we will explore identifying multiple covers.

References

1. Baader, F.: The description logic handbook: theory, implementation, and applications. Cambridge University Press (2003)
2. Beth, E.W.: On Padoa's method in the theory of definition. Indagationes Mathematicae 15, 330 – 339 (1953)
3. Euzenat, J., Shvaiko, P.: Ontology Matching, Second Edition. Springer (2013)
4. Geleta, D., Payne, T.R., Tamma, V.: An Investigation of Definability in Ontology Alignment. In: The 20th International Conference on Knowledge Engineering and Knowledge Management (EKAW2016). Springer (2016), To Appear
5. Geleta, D., Payne, T.R., Tamma, V.: Computing Minimal Signature Coverage for Description Logic Ontologies. Tech. Rep. ULCS-16-004, University of Liverpool (2016)
6. Hoogland, E., et al.: Definability and interpolation: Model-theoretic investigations. Institute for Logic, Language and Computation (2001)
7. Horridge, M., Bechhofer, S.: The OWL API: A Java API for OWL ontologies. Semantic Web 2(1), 11–21 (2011)
8. Jiménez-Ruiz, E., Payne, T.R., Solimando, A., Tamma, V.: Avoiding alignment-based conservativity violations through dialogue. In: Proc. OWLED. vol. 15 (2015)
9. Payne, T.R., Tamma, V.: Using Preferences in Negotiations over Ontological Correspondences. In: PRIMA 2015: Principles and Practice of Multi-Agent Systems, pp. 319–334. Springer (2015)
10. Santos, G., Tamma, V., Payne, T.R., Grasso, F.: Dialogue Based Meaning Negotiation. In: The 15th Workshop on Computational Models of Natural Argument (CMNA 2015) (2015)
11. Ten Cate, B., Franconi, E., Seylan, I.: Beth Definability in Expressive Description Logics. Journal of Artificial Intelligence Research (JAIR) 48, 347–414 (2013)
12. Van Harmelen, F., Ten Teije, A., Wache, H.: Knowledge engineering rediscovered: Towards reasoning patterns for the semantic web. In: Foundations for the Web of Information and Services, pp. 57–75. Springer (2011)
13. Vardi, M.Y.: Fundamentals of dependency theory. IBM Thomas J. Watson Research Division (1985)
14. Vazirani, V.V.: Approximation algorithms. Springer Science & Business M. (2013)