
Reasoning by Analogy Using Past Experiences

F. Leuzzi¹ and S. Ferilli^{1,2}

¹ Dipartimento di Informatica – Università di Bari
{fabio.leuzzi, stefano.ferilli}@uniba.it

² Centro Interdipartimentale per la Logica e sue Applicazioni – Università di Bari

Abstract. Reasoning by analogy is essential to provide new conclusions helpful to solve a problem. Here we present the definition of a new operator aimed at reasoning by analogy. The proposed reasoner relies on the *Roles Mapping Engine*. It finds analogous roles encoded in descriptions that use domain-specific terminology, overcoming syntactical constraints that limit the relations to have the same name. We employ also a structural similarity function to face cases affected by ambiguity. We evaluate our approach using examples proposed in other works, producing comparable results.

1 Introduction

As pointed out by [6], reasoning by analogy is essential in order to produce new conclusions helpful to solve a problem. In particular, some perspectives make this type of reasoning a primary issue. First, in the study of learning, analogies are important in the transfer of knowledge and inferences across different concepts, situations, or domains. Second, analogies are often used in problem solving and reasoning. Third, analogies can serve as mental models to understand new domains. Fourth, analogy is important in creativity. Studies in history science show that analogy was a frequent mode of thought for such great scientists as Faraday, Maxwell, and Kepler. Fifth, analogy is used in communication and persuasion. Sixth, analogy and its cousin, similarity, underlie many other cognitive processes. [6] defines the analogies as partial similarities between different situations that support further inferences. Specifically, analogy is defined as a kind of similarity in which the same system of relations holds across different objects. Analogies thus capture parallels across different situations.

This proposal consists in the definition of a new operator aimed at reasoning by analogy. The reasoner relies on the *Roles Mapping Engine* (RME), that finds common roles across descriptions. The long term objective is to embed in such definition a strategy for the cooperation with other reasoning operators.

The remainder of this work is organized as follows: in Section 2 related works are presented with related criticisms, in Section 3 some considerations about the analogy process are reported, then we present the proposed mapping procedure in details, presenting an evaluation in the successive section, finally we conclude with some considerations and future works.

2 Related Work

Plenty of works studied an operator aimed to perform reasoning by analogy. The most popular line of thought regards the research of identities between predicates across domain, using as representation formalism propositional or first order logic. In particular, in [13] the author aims to compose goal and sub-goal using analogous experiences. The authors claim that retrieving one (or many) analogies consists in finding *similar* past cases, then imposing to find the same predicate in both the experiences. This proposal contrasts with the canonical definition of analogy, in which only a correlation between roles is expected.

Similar assumption can be found in [10], in which the author proposes a strategy of knowledge projection between domains. This procedure is based on a definition of analogy presented in [9], that relies on the assumption that some given terms are analogous across domains if they are tied by the same predicate (as stated also in [8, 11]). In [5, 3, 7], the central statement is not so different: the author claims that analogy is characterized by the mapping of relations (having same predicate name) between objects, rather than attributes of objects, from base to target. The particular contribution that is slightly different from the other works is that this work isolates four primary classes of matching: literal similarity (in which both predicates and attributes are mapped), analogy (in which mainly predicates are mapped), abstraction (having an analogy like behaviour, but aimed to use the mapping for different goals) and anomaly (that is a wrong trying of analogical mapping). The current class depends on the domain in which an analogy is sought.

Again the same assumption is kept into account in [12], in which the author trains a classifier with a set of word pairs having as label the name of their relationship. The output of its algorithm is a classification, where the learned class describes a given relationship. All the pairs that are part of this class are claimed as analogues. We point out some limitations. In first place, the evaluation of a single relationship for each time is equivalent to consider each relationship in a complex context as independent from the others. In second place, this work proposes a supervised approach that require a concept description with a fixed list of features, such requirement is not always available in real cases.

A different approach has been proposed in [1], in which analogies are carried out using Evolutionary Computation. The authors represent the knowledge in semantic graphs and generate the dataset using common sense knowledge bases. At this point potential analogies are generated through evolution (i.e. using pairs of descriptions as parents probabilistically selected from population with reselection allowed) and evaluated through the Structure Mapping Engine (SME, [7]) that provides a fitness measure score. Despite the novelty of this approach, the methodology relies on the mapping of equal relation only.

In [2], the authors use the SME [7] in order to recognize similar sketches. In particular, they propose a software system in which an expert can draw a sketch that is stored as ground truth. In such a way a student can draw a sketch in turn, that is compared to the stored one with the aim to catch analogical aspects, checking its correctness. Unfortunately, this is a typical task of pattern

recognition, that fits with similarity evaluations. In fact, this work proposes an algorithm that exploits a numerical technique to revise the SME mistakes, instead of considering that analogy involves semantic aspects of reasoning that are far from pattern recognition in sketches.

In a general view, the attempt is to produce analogies searching identities between predicates across domains, we must underline that, despite the reliability of these works (shown by the experimental results), the research of identical predicates alone could be not enough to generate useful analogies.

3 The analogical reasoner

In this Section we will present our approach. For the sake of clarity, the description used as prior knowledge is referred as *base* domain, whereas the description of the current problem is referred as *target* domain.

We introduce some assumptions that limit our scopes: (1) we assume that all the descriptions are encoded using the same abstraction level, (2) this work does not keep into account the formalization of the goal, suggesting then all plausible analogies.

Retrieving oldest useful knowledge We want to face the retrieval of potentially analogous experiences. A good starting point can be an evaluation of common knowledge, that provides hints about potential points of contact between descriptions. Furthermore, the evaluation of the subnet of common knowledge relations allows to include the direct dependences between common statements. The addition of relational dimension allows the soundness check of the results.

In [11], the authors propose to face this step using a structural similarity evaluation, because they hope that this choice would increase the possibility of retrieval surprising and creative source domains. They describe the experiences in a n -dimensional structure space, where each dimension represents a particular topological quality of that domain. In such a way they can project the experiences in a vector space. Unfortunately, using such representation formalism some information are lost (e.g. connections among concepts).

For this reason, we decided to evaluate the similarity using the structural measure proposed in [4] and denoted as *fs*, because it performs a multi-level evaluation combining the similarities at terms, literals and clauses level. We apply the *fs* measure between a given description and each other one in the background knowledge. Unfortunately, such measure presents a drawback. Suppose given two short similar descriptions. In such a case the *fs* score will indicate similarity. Despite its score, the knowledge that can be effectively used for inferences could be poor. For this reason, such score is smoothed by a *multiplicative factor* denoted as *mf*. Given a set of clauses S , two clauses $C' \in S$ and $C'' \in S$, *mf* is computed as:

$$mf(C', C'') = \frac{|C'| + |C''|}{2 * L} = \frac{\mu(|C'|, |C''|)}{L}$$

Algorithm 1 Roles Mapping Engine.**Input:** A pair of Horn clauses $\langle C', C'' \rangle$.**Output:** Two sets: term mappings θ_t and predicate mappings θ_p .

```

 $\theta_p = \emptyset$ 
 $H = \langle h', h'' \rangle \mid h'$  is the head of  $C'$ ,  $h''$  is the head of  $C''$ 
 $\theta_t \leftarrow \theta_t \cup \{\text{term mappings in } H\}$ 
 $RA = \text{literals having the same predicate in } C' \text{ and } C''$ 
 $\theta_t \leftarrow \theta_t \cup \{\text{term mappings in } RA\}$ 
 $\theta_p \leftarrow \theta_p \cup \{\text{predicate mappings in } RA\}$ 
repeat
  repeat
     $\theta_p^{init} \leftarrow \theta_p, \theta_t^{init} \leftarrow \theta_t$ 
    repeat
       $\theta_p^{prev} \leftarrow \theta_p, \theta_t^{prev} \leftarrow \theta_t$ 
       $RA = \text{literals having the same predicate, and some terms in } \theta_t$ 
       $\theta_t \leftarrow \theta_t \cup \{\text{term mappings in } RA\}$ 
       $\theta_p \leftarrow \theta_p \cup \{\text{predicate mappings in } RA\}$ 
    until  $\theta_p \neq \theta_p^{prev} \vee \theta_t \neq \theta_t^{prev}$ 
    repeat
       $\theta_p^{prev} \leftarrow \theta_p, \theta_t^{prev} \leftarrow \theta_t$ 
       $RA = \text{literals having different predicates, and all terms in } \theta_t$ 
       $RRA = RA$  ranked by reliability score
       $\theta_p \leftarrow \theta_p \cup \{\text{predicate mappings in } RRA\}$ 
       $RA = \text{literals having different predicates, and some terms in } \theta_t$ 
       $RRA = RA$  ranked by reliability score
       $\theta_t \leftarrow \theta_t \cup \{\text{term mappings in } RRA\}$ 
    until  $\theta_p \neq \theta_p^{prev} \vee \theta_t \neq \theta_t^{prev}$ 
  until  $\theta_p \neq \theta_p^{init} \vee \theta_t \neq \theta_t^{init}$ 
   $\theta_p^{score} \leftarrow \theta_p, \theta_t^{score} \leftarrow \theta_t$ 
   $S = \langle l', l'' \rangle \mid l' \in C', l'' \in C'', S \text{ is } \arg \max_{(l_1, l_2) \in C' \times C''} rs(l_1, l_2), l' \text{ and } l'' \text{ are not}$ 
   $\text{fully mapped}$ 
   $\theta_t \leftarrow \theta_t \cup \{\text{term mappings in } S\}$ 
   $\theta_p \leftarrow \theta_p \cup \{\text{predicate mappings in } S\}$ 
until  $\theta_p \neq \theta_p^{score} \vee \theta_t \neq \theta_t^{score}$ 

```

where: $RA = \{ \langle \{L' \mid L' \subseteq C'\}, \{L'' \mid L'' \subseteq C''\} \rangle, \dots \}$ is a set of pairs of sets in which new mappings are sought; RRA is the list of sets in RA ranked by reliability score.

where $L = \arg \max_{C \in S} |C|$. Such a trick could avoid obvious or useless analogies between too short or too unbalanced descriptions.

Roles Mapping Engine Reasoning by analogy cannot be reduced to looking for equal predicates across domains, because the derived inference could be useless or trivial. In this section our mapping strategy is presented through the RME.

Each concept is represented in Horn clause logic. Telling our strategy in a nutshell, starting points are sought, then the mapping is expanded in breadth. Both steps are executed keeping consistency requirements.

We will discuss the RME approach using the Algorithm 1. In particular, such algorithm presents iterations on three levels: the external level has an iteration enclosing the whole life cycle of the analogical mapping; the middle level has an

iteration that checks whether one of the inner researches for mappings of identical and non-identical predicates produce novel mappings or not; the internal level has an iteration which aims to search for mappings of identical predicates and another one aimed to research mappings of non-identical predicates.

Such phases will be described in a formal notation and they will be equipped with a running example. Such example refers to the analogy between a proverb stating that “when the fox cannot reach the grapes, he says they are not ripe.” and a life context in which “a man cannot have a girl, he spreads a bad opinion about her”. We propose such example in order to clarify each step of our proposal.

Example 1 (Proverb and life context).

proverb(fox, grape) :- wants(fox, grape), cannot_take(fox, grape, fox_does_not_reach_grape), is(fox, crafty), cause(fox_does_not_reach_grape, bad_opinion), says(fox, grape_is_not_ripe), is(grape, not_ripe, grape_is_not_ripe), have(john, bad_opinion).

situation(john, carla) :- loves(john, carla), cannot_have(john, carla, john_cannot_have_carla), says(john, carla_is_bad), is(carla, bad, carla_is_bad), uses(jealous, craftiness).

In order to understand the mapping procedure, we need to define what are starting points and how to map terms and predicates as well.

Definition 1 (Starting point). *Given two clauses C' and C'' , a starting point S is a binding $S = [e'/e'']$ where $(e' \in \text{predicates}(C') \wedge e'' \in \text{predicates}(C'')) \vee (e' \in \text{terms}(C') \wedge e'' \in \text{terms}(C''))$.*

Definition 2 (Term mapping). *Given two clauses C' and C'' , two sets of literals $L' \subseteq C'$ and $L'' \subseteq C''$, a pair of terms t' and t'' , a consistent term association $\theta \subseteq \text{terms}(C') \times \text{terms}(C'')$; then t'/t'' is a term mapping if*

- $\{t'/t''\} \cup \theta$ is a consistent term association;
- either $\forall l' \in L'$ s.t. t' is a term of l' , $\forall l'' \in L''$ s.t. t'' is a term of l'' , both t' and t'' have position p ; or $\exists l', l'' \in L' \times L''$ s.t. t' is a term of l' , t'' is a term of l'' and both t' and t'' have name n and position p .

A term association θ is said to be consistent if it is a bijection, inconsistent otherwise.

Definition 3 (Predicate mapping). *Given two clauses C' and C'' , two sets of literals $L' \subseteq C'$ and $L'' \subseteq C''$ where $\forall l' \in L'$, $l' = p'(t'_1, \dots, t'_a)$ and $\forall l'' \in L''$, $l'' = p''(t''_1, \dots, t''_a)$, a consistent predicate association $\theta_p \subseteq \text{predicates}(C') \times \text{predicates}(C'')$, a consistent term association $\theta_t \subseteq \text{terms}(C') \times \text{terms}(C'')$; then p'/p'' is a predicate mapping if*

- $\{p'/p''\} \cup \theta_p$ is a consistent predicate association;
- either $p' = p''$; or a matching association defined as $\theta_{l'/l''} = \{t'_1/t''_1, \dots, t'_a/t''_a\}$ s.t. $\forall i = 1, \dots, a$, t'_i/t''_i is a term mapping, holds (i.e. $\theta_{l'/l''} \subseteq \theta_t$).

A predicate association θ_p is said to be consistent if it is a bijection, inconsistent otherwise.

Definition 4 (Compatibility under term and predicate mapping). *Two term mappings θ' and θ'' are t-compatible iff $\theta' \cup \theta''$ is consistent. Two literals, or two sequences of literals, are p-compatible iff all their predicate mappings are defined and consistent.*

Suppose given two clauses $C' = p'(t'_1, \dots, t'_n) :- l'_1, \dots, l'_k$ and $C'' = p''(t''_1, \dots, t''_m) :- l''_1, \dots, l''_h$ that the Algorithm 1 takes in input. Initially, we have an empty global term mapping θ_t and an empty global predicate mapping θ_p .

The heads mapping could be a good starting point. Since an analogy is sought between the descriptions of the concepts represented by the heads, this choice reflects the intrinsic strength of the heads relationship. More formally, if $n = m$, then $\theta'_t = \{t'_1/t''_1, \dots, t'_{n=m}/t''_{n=m}\}$ is a term mapping. Since θ_t is empty, θ_t and θ'_t are t-compatible, then $\theta_t \leftarrow \theta_t \cup \theta'_t$.

Although the entities in the descriptions will play specific roles (likely using a domain specific terminology), it is plausible that part of explanations are encoded using common sense, providing potential points of contact. Then a research of equal predicates across descriptions makes sense. The underlying idea is that two domains in the same knowledge share the representation formalism (implying the common sense as intersection between any pair of descriptions).

The first iteration of the internal level in the Algorithm 1 can be formalized as follow. Using Definitions 2, 3 and 4, we carry on our mappings searching for shared knowledge. Given two subsets $L' \subseteq C'$ and $L'' \subseteq C''$ s.t. $\forall l' \in L', l' = p(t'_1, \dots, t'_a)$ and $\forall l'' \in L'', l'' = p(t''_1, \dots, t''_a)$, then $\theta'_p = \{p/p\}$ is a predicate mapping. If θ_p and θ'_p are p-compatible, then $\theta_p \leftarrow \theta_p \cup \theta'_p$.

Example 2 (Proverb and life context). Firstly, the heads are mapped, since they have the same arity, obtaining $\langle \text{grape}, \text{carla} \rangle$, $\langle \text{fox}, \text{john} \rangle$. At this point the literals having a predicate used by both clauses are isolated, in order to search for deterministic alignment. Then we have $\{\text{says}(\text{fox}, \text{grape_is_not_ripe})\}$ and $\{\text{says}(\text{john}, \text{carla_is_bad})\}$, from which $\langle \text{says}/2, \text{says}/2 \rangle$ becomes a mapped predicate and $\langle \text{grape_is_not_ripe}, \text{carla_is_bad} \rangle$ becomes a mapped term. Going on, the sets $\{\text{is}(\text{grape}, \text{not_ripe}, \text{grape_is_not_ripe})\}$ and $\{\text{is}(\text{carla}, \text{bad}, \text{carla_is_bad})\}$ are isolated, in which some terms are mapped, and from which we can add to the mapped predicates $\langle \text{is}/3, \text{is}/3 \rangle$ and to the mapped terms $\langle \text{not_ripe}, \text{bad} \rangle$.

In Algorithm 1, the second iteration of the internal level tries to map non-identical predicates expanding in breadth the previous mappings to those concepts that are syntactically different but that play an analogous role. The research of predicates and terms association goes on until new mappings are done. More formally, suppose given two subsets of literals $L' \subseteq C'$ and $L'' \subseteq C''$ for which $\exists t'/t'' \in \theta_t$ s.t. $\forall l' \in L', t'$ is a term of l' and $\forall l'' \in L'', t''$ is a term of l'' . If exists a term mapping θ'_t s.t. $\forall t'/t'' \in \theta'_t, t'$ is a term of $l' \in L'$ in position w , t'' is a term of $l'' \in L''$ in position w , θ_t and θ'_t are t-compatible, then $\theta_t \leftarrow \theta_t \cup \theta'_t$. Moreover, if exists a predicate mapping θ'_p s.t. $\forall p'/p'' \in \theta'_p, \forall l' \in L', l' = p'(t'_1, \dots, t'_a), \forall l'' \in L'', l'' = p''(t''_1, \dots, t''_a), \forall i = 1, \dots, a$ then $\exists t'_i/t''_i \in \theta_t, \theta_p$ and θ'_p are p-compatible, then $\theta_p \leftarrow \theta_p \cup \theta'_p$.

Table 1. Mapping between proverb and life context.

Outcome	Base clause	Target clause
mapped predicates	says/2	says/2
	is/3	is/3
	wants/2	loves/2
	cannot_take/3	cannot_have/3
mapped terms	fox	john
	grape	carla
	grape_is_not_ripe	carla_is_bad
	not_ripe	bad
	fox_does_not_reach_grape	john_cannot_have_carla
	crafty	craftiness

Example 3 (Proverb and life context). In the Example 2 a set of mapped predicates and a set of mapped terms have been obtained. Since $\{ \langle \text{grape}, \text{carla} \rangle, \langle \text{fox}, \text{john} \rangle \} \subset \theta_t$ (see Algorithm 1), the expansion in breadth produces the pair of sets $\{ \text{wants}(\text{fox}, \text{grape}) \}$ and $\{ \text{loves}(\text{john}, \text{carla}) \}$, from which $\langle \text{wants}/2, \text{loves}/2 \rangle$ is added to θ_p . Consequently, $\{ \text{cannot_take}(\text{fox}, \text{grape}, \text{fox_does_not_reach_grape}) \}$ and $\{ \text{cannot_have}(\text{john}, \text{carla}, \text{john_cannot_have_carla}) \}$ is found, allowing to map $\langle \text{cannot_take}/3, \text{cannot_have}/3 \rangle$ that can be added to θ_p and $\langle \text{fox_does_not_reach_grape}, \text{john_cannot_have_carla} \rangle$ that can be added to θ_t .

The middle level iteration in Algorithm 1 ensures that the internal iterations are repeated until at least one of them extends the mappings. Then, the middle level performs all deterministic mappings based on previous ones.

Unfortunately, it could be the case in which common knowledge does not exist, or cannot be used as starting point, or deterministic mappings are not available. A strategy relying on the structure analysis becomes a primary issue, in order to suggest starting points that do not share the representation. Then a structural similarity is exploited in order to obtain a reliability score between literals (denoted as rs). In such a way we design a pair of literals as new starting point, if it is the most similar, it is not already mapped and its literals have the same arity. This is the reason for which the external level exists. It attempts to restart the mappings expansion using such evaluation to overcome the absence of deterministic mappings. This attempt can be seen as the last opportunity to make novel deterministic mappings after the end of the algorithm.

More formally, we obtain a pair $l'/l'' \in C' \times C''$ s.t. $l' = p'(t'_1, \dots, t'_a)$, $l'' = p''(t''_1, \dots, t''_a)$, $rs(l', l'')$ is the maximum score, l'/l'' has not been fully mapped through term and predicate mappings. Then $\theta'_t = \{ t'_1/t''_1, \dots, t'_a/t''_a \}$ and $\theta'_p = \{ p'/p'' \}$. If $\theta_t \cup \theta'_t$ is t-compatible, then $\theta_t \leftarrow \theta_t \cup \theta'_t$. If θ_p and θ'_p are p-compatible, then $\theta_p \leftarrow \theta_p \cup \theta'_p$.

Example 4 (Proverb and life context). At this point, the mapping expanded in breadth pursued in the Example 3 (see Table 1) cannot be carried on because there are not novel deterministic bindings. The similarity function suggests $\{ \text{is}(\text{fox}, \text{crafty}) \}$ and $\{ \text{uses}(\text{jealous}, \text{craftiness}) \}$. Here, only $\langle \text{crafty}, \text{craftiness} \rangle$

is a deterministic mapping. Since *fox* is already in θ_t as $\langle fox, john \rangle$, *jealous* cannot be mapped, impeding the association of the respective predicates.

Ranking of hypotheses A known problem is the ranking of the hypotheses to learn. Such problem affects reasoning by analogy, since each new mapping relies on previous ones, and so on. In this proposal, the hypotheses ranking problem has been faced using relative frequencies.

Given a clause C and a literal $l \in C$, the score of l is obtained averaging the relative frequencies for each term t in l , then:

$$rf_{\mu}(l) = \frac{\sum_{i=1}^a o_i}{n * a}$$

where o is the number of literals in C in which the t appears, n is the total number of literals in C , a is the arity of l . The idea behind the $rf_{\mu}(l)$ scores is to represent the centrality of l in its clause. Then, given two literals l' and l'' , the reliability score $rs(l', l'')$ for their hypothesis of mapping is computed as:

$$rs(l', l'') = rf_{\mu}(l') * rf_{\mu}(l'')$$

Mappings hypotheses are ranked using a descendant $rs(l', l'')$ score. A ranking so defined means that each new mapping maximizes the coverage of literals in both clauses.

Making inference During the mapping phase, the attention has been focused on the recognition of analogous roles cross-domains (both for objects and relations). As highlighted in [3], one-to-one alignment has a primary importance, since part of the structural consistency is verified if the bijection holds. For this reason, the inference is carried out starting from a one-to-one alignment of the mapped literals (i.e. both for the predicate and its arguments), and ending with the projection of all the residual knowledge.

Giving a more practical view of the procedure, after the filtering out of the one-to-one correspondences, the procedure seeks base knowledge that could be novel in the target domain. Consistently with the assumption that common knowledge is fundamental for analogical inference, it is possible that the lacking mappings are part of common knowledge, then it is projected using a Skolem function. An inference having Skolem functions is a hypothesis having an unreliability degree directly proportional to the number of Skolem functions.

Example 5 (Proverb and life context). The expected explanation of the phenomena sounds like: “John has a bad opinion about Carla because he cannot have the love of Carla, then he uses his craftiness in order to do not appear rejected from the girl”. Let us examine the inference hypotheses from the proverb to the life context:

1. *skolem_is(john, craftiness)*
2. *skolem_cause(john_cannot_have_carla, skolem_bad_opinion)*
3. *skolem_has(john, skolem_bad_opinion)*

These hypotheses fully satisfy the expected interpretation.

Table 2. Literal mappings between proverb and life context.

Proverb	Life context
proverb(fox, grape)	situation(john, carla)
says(fox, grape_is_not_ripe)	says(john, carla_is_bad)
is(grape, not_ripe, grape_is_not_ripe)	is(carla, bad, carla_is_bad)
wants(fox, grape)	loves(john, carla)
cannot_take(fox, grape,	cannot_have(john, carla,
fox_does_not_reach_grape)	john_cannot_have_carla)

Meta-Pattern formalization Using the RME we can carry out an analogical mapping between each pair of clauses representing experiences, contexts or concepts. From such a mapping we can outline a pattern.

Let us to give a formal view of a pattern. Given two clauses $C' = h' :- l'_1, \dots, l'_n$ and $C'' = h'' :- l''_1, \dots, l''_m$, a set θ_t containing the mapped terms and a set θ_p containing the mapped predicates, we can outline a generalized pattern $C = h :- l_1, \dots, l_k$ with $k \leq \min(n, m)$, s.t. $\forall l \in C, \exists l'/l''$ s.t. $l' \in C', l'' \in C'', l' = p'(t'_1, \dots, t'_a), l'' = p''(t''_1, \dots, t''_a), p'/p'' \in \theta_p$ and $t'_i/t''_i \in \theta_t$ ($\forall i = 1, \dots, a$). Moreover, $l = p(t_1, \dots, t_a)$ where: if $p' = p''$, then $p = p' = p''$, otherwise $p = p*$, where $p*$ is a new predicate; if $t'_i = t''_i$, then $t_i = t'_i = t''_i$, otherwise $t_i = t*_i$, where $t*_i$ is a new term.

Since each analogy has a reason to exist with respect to a specific perspective, we cannot expect that each pattern will become general. Conversely, very often analogies remain specific. Then each refinement of a pattern needs to be consider a new pattern, because we have not any reason to consider too specific or useless the older pattern. In any case, we can say that trying an analogical mapping between a pattern and a third description (or another pattern), if the pattern(s) is not fully mapped, a novel and more general *meta-pattern* arises.

The reason behind such a choice is that if the pattern is used to make a novel analogy, many domains can support a mapping or suggest the argument of the relative Skolem function. To note that for each use or refinement of a pattern, the novel domain contributes to support each survived mapping in the pattern, giving further confirmation that the mappings in which the name of the original predicate or term has been preserved are effectively common sense expressions.

The story of each predicate/term in the pattern can be recognized, since the origin of each predicate/term in the pattern is stored using the formalism:

$$db(head, type, pattern_name, original_name)$$

where ‘head’ stands for the head of the original clause, ‘type’ indicates if we are storing a predicate or a term, ‘pattern_name’ represents the name reported in the pattern and the ‘original_name’ reports the name in the original clause.

Example 6 (Proverb and life context). The mappings presented in Table 1 bring to the literals alignment in Table 2, from which we obtain the following pattern.

$pattern(proverb(fox, grape), situation(john, carla)) :-$
wants_OR_loves(fox_OR_john, grape_OR_carla),

```
cannot_take_OR_cannot_have(fox_OR_john, grape_OR_carla,
    fox_does_not_reach_grape_OR_john_cannot_have_carla),
says(fox_OR_john, grape_is_not_ripe_OR_carla_is_bad),
is(grape_OR_carla, not_ripe_OR_bad, grape_is_not_ripe_OR_carla_is_bad).
```

What is an analogy? In order to provide a formal definition of analogy, we need to formalize what is the role that an object plays in a description. Keeping in mind that an object is an abstract entity that can be materialized differently in each description, let us to define a role.

Definition 5 (Role). *Given a clause C , and a literal $l = p(t_1, \dots, t_a)$ s.t. $l \in C$, then the role of a term $t_i (1 \leq i \leq a)$ is a set $R_{t_i} = \{l_1, \dots, l_k\}$ s.t. $\forall l' \in R_{t_i}, l' \in C$ and t_i appears in l' . Such a set refers to the role that the term t_i plays with respect to the other terms involved in the relations in which it is involved.*

In any situation a task can be recognized. Any task requests a focus, i.e. the set of objects and relations necessary and sufficient to carry out the current task. Sometimes objects or relations lack, then an analogy could represent a way to hypothesize them from previous experiences. We need to retrieve the experience having an alignable focus, in order to derive hypotheses. In the light of such premises, we define the analogical perspective.

Definition 6 (Analogical perspective). *Given two descriptions $\langle D', D'' \rangle$ representing respectively base and target domain, K' and K'' denote the aligned knowledge among D' and D'' , T' and T'' denote the aligned knowledge among D' and D'' that solves the current task, an analogical perspective holds if either*

$$T' \subseteq K' \subseteq D' \wedge T'' \subseteq K'' \subseteq D''$$

or

$$T' \subseteq K' \subseteq D' \wedge T'' \subseteq (K'' \cup I) \subseteq D''$$

Where I is the inference obtained from the base domain.

Definition 7 (Analogy). *Given two relational descriptions $\langle D', D'' \rangle$ representable as sets of roles $\langle R_{D'}, R_{D''} \rangle$ and a perspective P , we say that D' and D'' are analogous if there exist two subsets $S_{D'} \subseteq R_{D'}$ and $S_{D''} \subseteq R_{D''}$ s.t. a bijective function $f : S_{D'} \rightarrow S_{D''}$ holds, and f satisfies P . f satisfies P if for each role $r \in R_{D'} \vee R_{D''}$ necessary to explain P , f holds (i.e., $r \in S_{D'} \vee S_{D''}$).*

Remark 1 (Analogy). Given a description, each object plays a specific role with respect to each other. Common roles across descriptions are essential to the analogy, relations analysis is fundamental for roles identification, common objects can be a clue for relations analysis.

4 Evaluation

We present a qualitative evaluation, in such a way we can further clarify our proposal. In [8] the analogical reasoning is explored from the cognitive psychology perspective. The authors want to study the analogical process of reasoning in humans. Then they give to a group of humans two stories: the former talks about a general that wants to capture a fortress, whereas the latter talks about a doctor that wants to defeat a tumor. These stories are respectively the base and the target domain. The human subjects completed the latter story in the light of the former one (i.e. trying to recognize the knowledge that solves the problem in the target story). Without any suggestion, only the 57% of the subjects provided a complete solution to the analogy, whereas our software system implementing the RME provides directly the correct analogical solution.

RME assessment Let us give details about the stories and the relative expected solution. The base story follows.

A fortress was located in the center of the country. Many roads radiated out from the fortress. A general wanted to capture the fortress with his army. The general wanted to prevent mines on the roads from destroying his army and neighbouring villages. As a result the entire army could not attack the fortress along one road. However, the entire army was needed to capture the fortress. So an attack by one small group would not succeed. The general therefore divided his army into several small groups. He positioned the small groups at the heads of different roads. The small groups simultaneously converged on the fortress. In this way the army captured the fortress.

The base story is translated in a Horn clause having as head *conquer(fortress)*. Each item in the following list encodes a sentence in the story.

1. *located(fortress,center), partof(center,country),*
2. *radiated(oneroad,fortress), radiated(roads,fortress), partof(oneroad,roads),*
3. *capture(general,fortress), use(general,army),*
4. *prevent(general,mines), located(mines,oneroad), located(mines,roads),*
destroy(mines,army), destroy(mines,villages),
5. *couldnotuse(army,oneroad),*
6. *capture(army,fortress),*
7. *couldnotuse(subgroup,oneroad),*
8. *splittable(army,subgroups), partof(subgroup,subgroups), partof(subgroups,army),*
destroy(mines,subgroup), notenough(subgroup),
9. *distribute(subgroups,roads),*
10. *converge(subgroups,fortress),*
11. *capture(subgroups,fortress).*

The target story follows.

A tumor was located in the interior of a patient's body. A doctor wanted to destroy the tumor with rays. The doctor wanted to prevent the rays from destroying healthy tissue. As a result the high-intensity rays could not be applied to the tumor along one path. However, high-intensity rays were needed to destroy the tumor. So applying one low-intensity ray would not succeed.

Table 3. Military and medical mapping outcomes.

Outcome	Base clause	Target clause	Outcome	Base clause	Target clause
mapped predicates	destroy/2	aredestroyed/2	mapped arguments	country	body
	capture/2	defeat/2		center	interior
	partof/2	partof/2		roads	slits
	couldnotuse/2	couldnotuse/2		subgroups	subrays
	splittable/2	splittable/2		oneroad	oneslit
	use/2	use/2		army	rays
	radiated/2	radiated/2		mines	healthytissue
	prevent/2	prevent/2		general	doc
	located/2	located/2		subgroup	ray
	notenough/1	notenough/1		fortress	tumor

The target story becomes a Horn clause having the head $heal(tumor)$. The last item encodes implicit knowledge. In particular such item says that the cancer can be reached from one or many directions. It encodes also that a slit is one of many slits, that the rays can be splitted and that healthy tissue can be damaged and/or destroyed from rays or sub-rays.

1. $located(tumor, interior), partof(interior, body),$
2. $defeat(doc, tumor), use(doc, rays),$
3. $prevent(doc, healthytissue), located(healthytissue, oneslit),$
 $located(healthytissue, slits), aredestroyed(healthytissue, rays),$
4. $couldnotuse(rays, oneslit),$
5. $defeat(rays, tumor),$
6. $couldnotuse(ray, oneslit),$
7. (additional) $radiated(oneslit, tumor), radiated(slits, tumor),$
 $partof(oneslit, slits), splittable(rays, ray), partof(ray, subrays),$
 $partof(subrays, rays), aredestroyed(healthytissue, ray),$
 $aredestroyed(healthytissue, subrays), notenough(ray).$

The expected result must contain the lacking knowledge of the target story, that is: “The doctor therefore divided the rays into several low-intensity rays. He positioned the low-intensity rays at multiple locations around the patient’s body. The low intensity rays simultaneously converged on the tumor. In this way the rays destroyed the tumor.”

Given the mapping in Table 3, the inference hypotheses from base to target domain are:

1. $defeat(subrays, tumor)$
2. $splittable(rays, subrays)$
3. $skolem_distribute(subrays, slits)$
4. $skolem_converge(subrays, tumor)$
5. $aredestroyed(healthytissue, skolem_villages)$

The hypothesis 1 can be identified as the goal of the problem, it is made of fully mapped components, making it a conclusive inference. The hypotheses 2, 3 and 4 represent the procedure useful to reach the goal, their predicates are

Table 4. Pattern and Pharmaceutical mapping outcomes.

Outcome	Pattern	Target clause
mapped predicates	capture_OR_defeat/2	purchase/2
	partof/2	partof/2
	couldnotuse/2	couldnotuse/2
	located/2	located/2
	use/2	use/2
mapped terms	prevent/2	mustface/2
	country_OR_body	pharmacy
	subgroups_OR_subrays	many_partial_amounts
	subgroup_OR_ray	partial_amount
	center_OR_interior	warehouse
	oneroad_OR_oneslit	one_money_source
	army_OR_rays	medicine_total_amount
	mines_OR_healthytissue	not_enough_money
general_OR_doc	patient	
fortress_OR_tumor	medicine	

not mapped, then *skolem* has been added (i.e. a Skolem function), in order to emphasize that the relation could be common sense knowledge, making it projectable without further elaborations. In any case, this type of inference needs to be considered contingent instead of conclusive. The hypothesis 5 does not make sense, then it is a case of fake inference. It is straightforward to highlight that all these statements was absent in the target domain, and have been completely obtained using the base domain. Finally, it is easy to note the consistency with the expected analogical reasoning.

In order to evaluate the similarity of the relational structure, the similarity between the clauses has been evaluated using the *fs* measure [4] before and after the use of the RME. The *fs* ranges between]0,1[. The original clauses score is 0.65, whereas after alignments, the score became 0.85. The alignment allowed the recognition of the 20% of the structure. Such portion of the clauses appeared unrelated before the RME.

Patterns assessment The proposed approach has been evaluated using a qualitative experiment that carries on the running example of analogical reasoning between military and medical domains, that produced a pattern as described in Section 3. We chosen a third domain telling about the purchase of a medicine for which an offertory is needed. We refer to this story as *Pharmaceutical*.

A medicine is located in the warehouse of a pharmacy. A patient needs to purchase the medicine. The patient must face the problem that his money is not enough. As a result the patient cannot purchase the medicine paying the total price. However, the total amount is needed to purchase the medicine. So applying a minor amount cannot succeed.

The clause encoding such concepts has the head *get(medicine)*.

Table 5. Suggestions from original domains.

Pharmaceutical	Mapping	Source
mustface/2	prevent/2	Military Medical
use/2	use/2	Military Medical
located/2	located/2	Military Medical
couldnotuse/2	couldnotuse/2	Military Medical
partof/2	partof/2	Military Medical
purchase/2	capture/2 defeat/2	Military Medical
medicine	fortress tumor	Military Medical

Pharmaceutical	Mapping	Source
patient	general doc	Military Medical
not_enough_money	mines healthyissue	Military Medical
medicine_total_amount	army rays	Military Medical
one_money_source	oneroad oneslit	Military Medical
warehouse	center interior	Military Medical
partial_amount	subgroup ray	Military Medical
many_partial_amounts	subgroups subrays	Military Medical
pharmacy	country body	Military Medical

1. *located(medicine, warehouse), located(warehouse, pharmacy),*
2. *purchase(patient, medicine), use(patient, medicinetotalamount),*
3. *mustface(patient, notenoughmoney),*
4. *couldnotuse(medicinetotalamount, onemoneysource),*
5. *couldnotuse(partialamount, onemoneysource),*
6. *purchase(medicinetotalamount, medicine),*
7. (additional) *splittable(medicinetotalamount, partialamount), partof(partialamount, manypartialamounts), partof(manypartialamounts, medicinetotalamount).*

In the light of the mappings in Table 4, the RME suggests analogous domains using the stored original mappings, reported in Table 5. For the predicates *use/2*, *located/2*, *couldnotuse/2* and *partof/2* there is no novelty. Instead, the predicates *mustface/2* and *purchase/2* are more interesting because the suggestion indicates that *mustface/2* is the “difficulty” that the protagonist must solve, whereas *purchase/2* stands for the main action on which Pharmaceutical story is built.

The term mapping suggestions have not shared knowledge, then each term is traced to different terms in base domains. For instance *patient* is traced to the main actors in the other domains (*general* and *doc*), such as *medicine* that is the target object in the story is traced to *fortress* and *tumor*, and so on.

As for the analogy between Military and Medical domains, also here the *fs* measure has been exploited to evaluate the gain in the alignment of the portion of the structure for which the analogy holds. The original clauses score is 0.4, whereas the score after the RME is 0.74. Then the 34% of the structure that appeared not related in the original clauses, has been aligned in order to bring out the analogy.

The evaluation of the inference step is important in turn. Since the projection of knowledge is not empty, we can conclude that the analogy with Pharmaceutical domain is well represented by another pattern. The consequence is that a novel pattern has been produced.

5 Conclusions

In this work we propose a strategy aimed to recognize analogies and to build meta-patterns for further reasoning, generalizing the analogical schemas. Our proposal differs from the existing literature since it allows to learn patterns representing both the intuition to know any potential solution to the problem, and a computational trick that allows to reuse analogies computed in the past. Moreover, the RME captures non-syntactic alignments without meta-descriptions.

Future improvements will regard relations with opposite sense (perhaps using a common sense knowledge). Another interesting direction could be the use of a probabilistic approach to mappings of non identical predicates. We plan also to face the factual validity using the abductive procedure, in order to check if the inferred knowledge (mapped or projected) is consistent with the constraints of the world. Last but not least, we will equip the solution with an abstraction operator, in order to shift the representation if needed.

References

- [1] Atilim Günes Baydin, Ramon López de Mántaras, and Santiago Ontañón. Automated generation of cross-domain analogies via evolutionary computation. *CoRR*, abs/1204.2335, 2012.
- [2] Maria de los Angeles Chang and Kenneth D. Forbus. Using quantitative information to improve analogical matching between sketches. In *IAAI*, 2012.
- [3] Brian Falkenhainer, Kenneth D. Forbus, and Dedre Gentner. The structure-mapping engine: Algorithm and examples. *Artificial Intelligence*, 41:1–63, 1989.
- [4] S. Ferilli, M. Biba, N. Di Mauro, T.M. Basile, and F. Esposito. Plugging taxonomic similarity in first-order logic horn clauses comparison. In *Emergent Perspectives in Artificial Intelligence*, Lecture Notes in Artificial Intelligence, pages 131–140. Springer, 2009.
- [5] Dedre Gentner. Structure-mapping: A theoretical framework for analogy. *Cognitive Science*, 7(2):155–170, 1983.
- [6] Dedre Gentner. Analogy. *A companion to cognitive science*, pages 107–113, 1998.
- [7] Dedre Gentner and Arthur B. Markman. Structure mapping in analogy and similarity. *American psychologist*, 52:45–56, 1997.
- [8] Mary L. Gick and Keith J. Holyoak. Analogical problem solving. *Cognitive Psychology*, 12(3):306–355, 1980.
- [9] Makoto Haraguchi. Towards a mathematical theory of analogy. *Bulletin of informatics and cybernetics*, 21(3):29–56, 1985.
- [10] Makoto Haraguchi. Analogical reasoning using transformations of rules. In *Proceedings of the 4th conference on Logic programming '85*, pages 56–65, New York, NY, USA, 1986. Springer-Verlag New York, Inc.
- [11] Diarmuid P. O'Donoghue and Mark T. Keane. A creative analogy machine: Results and challenges. In *Proceedings of the International Conference on Computational Creativity 2012*, 2012.
- [12] Peter D. Turney. A uniform approach to analogies, synonyms, antonyms, and associations. *CoRR*, abs/0809.0124, 2008.
- [13] Manuela M. Veloso and Jaime G. Carbonell. Derivational analogy in prodigy: Automating case acquisition, storage, and utilization. In *Machine Learning*, pages 249–278. Kluwer Academic Publishers, Boston, 1993.