The Role of Analogy in Ontology Alignment: A Study on LISA

Elie Raad[a], Joerg Evermann[a,*]

[a]*Faculty of Business, Memorial University of Newfoundland, St. John's, NL, Canada*

Abstract

Ontologies are explicit specifications of concepts and their relationships. In the context of a semantic web of independently developed ontologies, overcoming interoperability and heterogeneity issues is of considerable importance. Many semantic web applications, such as matching of instances in social networks, reasoning over combined knowledge bases, and knowledge sharing among services, rely on ontology alignment. While existing research in this area has developed a wide range of different heuristics, in this paper we propose to look towards cognitive science, specifically analogical reasoning, to support ontology alignment. We investigate the question whether ontology alignment is rooted in the same cognitive process as analogical reasoning. We apply the LISA system, a cognitively-based model of human analogical reasoning, to ontology alignment and present a comprehensive experimental study to determine its performance on ontology alignment problems.

Keywords: Ontology alignment, Analogy, Cognition, LISA

1. Introduction

Ontologies are explicit, formal representations of concepts and their relationships within a domain of knowledge (Gruber, 1993). They are used in a variety of research areas such as knowledge management (Sure et al., 2002), geographic information representation (Kuhn, 2001), medical information modeling (Pisanelli et al., 2000), user profile matching (Raad et al., 2010), and web mining (Aufaure et al., 2006). Ontologies are important to achieve interoperability among heterogeneous systems and are a key enabling technology for the semantic web vision (Berners-Lee et al., 2001). The distributed and independent nature of the semantic web makes it likely that multiple parties independently develop ontologies that partially overlap in the knowledge they represent. Ontology alignment methods attempt to establish correspondences between concepts of different ontologies.

Existing ontology alignment research has produced a wide range of methods, which include estimating lexical similarity between concepts (using syntactic or semantic similarity), identifying structural similarities between ontologies (using tree-based or graph-based representations), as well as relying on human interventions (assigning weights or confidence scores, setting alignment parameters, confirming alignment results, etc.) (Lambrix and Tan, 2006; Hong-Hai and Rahm, 2007; Wei et al., 2008; Cruz et al., 2009). Further, recent research has moved to include sets or ensembles of individual matchers, using heuristics to select, tune, and order the various matchers, weights, and thresholds within the overall system. Although these approaches are individually based on well-justified logical, linguistic and statistical ideas, their diversity suggests a lack of underlying theoretical foundation. We propose that human anological reasoning processes can provide such a theoretical foundation.

We argue that in order for an ontology alignment method to judged successful, the alignments it produces should match those generated by humans. Only then does the alignment method meet the expectations of its users, and only then will the system be accepted as successful. This argu-

*Corresponding author

Email addresses: `eliraad@gmail.com` (Elie Raad), `jevermann@mun.ca` (Joerg Evermann)

ment is reflected by the fact that ontology alignment systems are routinely evaluated against reference alignments created by humans (Evermann, 2008b). While this does not imply that a successful ontology alignment method must necessarily operate according to human cognitive principles, it does suggest that a method that does work according to such cognitive principles may be more likely to perform well.

In this paper, we follow recent calls to bridge the gap between cognitive science and the semantic web (Raubal and Adams, 2010; Gentner et al., 2012), and investigate whether and how existing computational models of analogical reasoning can be applied to the ontology alignment problem, a semantic web use case. While there is a great amount of research in the area of analogical reasoning and numerous computational models have been proposed, this paper focuses on the LISA model (Learning and Inference with Schemas and Analogies) (Hummel and Holyoak, 1997), a prominent and empirically well-validated computational analogical reasoning system.

Our approach of applying cognitive science research to semantic web problems is also supported by recent work on database schema matching, a problem similar to ontology alignment. That work has investigated how humans make schema matching decisions (Evermann, 2009, 2008a,b, 2010) with the aim of applying the discovered principles and methods to the matching problem. However, in contrast to the present work, that work was grounded in theories of meaning rather than cognitive analogical reasoning. More recently, cognitive principles of similarity have been applied to the schema matching problem (Lukyanenko and Evermann, 2011; Evermann, 2012), but so far only with a brief demonstration of principles.

The remainder of the paper is structured as follows. Section 2 briefly introduces the cognitive science research on analogical reasoning systems, presents LISA, the analogical reasoning systems adopted in this research, and describes our approach to ontology alignment with LISA. Section 3 describes our experimental design and results, including a comparison with existing ontology alignment approaches. Section 4 discusses the contributions of the study, and suggests several directions for future research.

2. Analogical Reasoning with LISA

Analogical reasoning is a cognitive process that is used for learning from prior knowledge. It maps knowledge from one domain (the base) into another domain (the target) such that relations that hold among the base objects also hold among the target objects (Gentner, 1983). Analogies in human cognition have been shown to follow three main principles (Gentner and Forbus, 2011):

1. *One-to-one mapping:* Humans are shown to construct and to strongly prefer analogies that place each concept of the base domain in correspondence with at most one object of the target domain.
2. *Systematicity:* Analogies become more useful when they are able to map objects based on the objects' structure rather than based on superficial similarities (Gentner, 1983). For example, in the analogy "an electric battery is like a reservoir", there is no resemblance between the objects' surface attributes (e.g., shape, size, color). However, the analogy builds on the fact that both objects store potential energy, i.e. it is based on structural similarity with a common "higher-order" relation *stores energy.*
3. *Parallel connectivity:* This principle states that if predicates of propositions are mapped to each other, their constituent objects must also be placed in correspondence. For example, if the predicate *stores* in the proposition *stores (reservoir, energy)* is matched to the predicate *stores'* in *stores' (electric battery, energy')* then *reservoir* must match *electric battery* and *energy* must match *energy'.*

Analogies are at the core of human cognition (Gust et al., 2008; Gick and Holyoak, 1980; Gentner and Forbus, 2011) and are a promising field of research in cognitive science (Forbus et al., 1998). Reasoning by analogy has received significant attention in cognitive systems research (French, 2002; Gentner, 2003; Krawczyk et al., 2004; Morrison et al., 2011). A number of computational models of analogy-making systems have been proposed, implemented (Falkenhainer et al., 1989; Goldstone and Medin, 1994; Gust et al., 2006; Hummel and Holyoak, 1997) and experimentally evaluated (Forbus, 2001; Gentner, 2010; Gentner and Forbus, 2011; Loewenstein and Gentner, 2005; Lovett et al., 2009). While SME (Structure-Mapping-Engine) (Falkenhainer et al., 1989) and LISA (Learning and

Inference using Schemas and Analogies) (Hummel and Holyoak, 1997) have had the greatest success in accounting for the range of phenomena in analogical thinking and learning (Gentner, 2010), we focus on LISA as it does not make the assumption of common relationships names that SME does. Moreover, LISA's architecture is based on cognitively realistic assumptions about working memory capacity and recall. Thus, it is likely that LISA more accurately reflects human processes to generate object and relational mappings (Krawczyk et al., 2004). Hummel and colleagues have explored the performance of LISA for retrieving and mapping analogs (Hummel and Holyoak, 1997) and for inferring and inducing schemas (Hummel and Holyoak, 2003). LISA was also successfully applied to modeling the loss of relational reasoning in populations with forms of brain damage (Morrison et al., 2004), studying children's development of analogical reasoning (Richland et al., 2006), and comparing the reasoning performance between young and old adults (Viskontas et al., 2004).

2.1. Symbolic Knowledge Representation

LISA is based on a combination of symbolic and connectionist architectures. A domain of knowledge is represented in LISA using units that represent logical propositions ("P units", closed atomic formulas or ground atoms in first-order logic). Each proposition is connected to a set of role units ("R" units). In first-order logic, these are free variables in the open formula corresponding to the proposition (the "names" of the arguments). Each proposition is also connected to a set of object units ("O" units). In first-order logic, these are the ground terms in the closed atom that are substituted for (bound to) the free variables in the open atomic formula corresponding to the proposition. The ground substitution of each free variable with a ground term is called a sub-proposition ("SP unit") in LISA terminology. Table 1 gives a summary and an example of this terminology.

Additionally, roles and objects in LISA are connected to semantic units. These have no equivalent in the corresponding first-order logic knowledge representation, but are literals that serve to characterize the roles and objects. For example, the concept of an academic paper, represented by the object unit (or ground term) *paper*, might be associated with literals such as "present", "conference", etc. Similarly, the concept of a subclass, represented by the role unit (or free variable) *SubClass*, might be associated with the literals "OWL" (indicating that it is part of the Web Ontology Language OWL), "sub-ordinate", "class" (indicating that it refers to a class in the ontology). These associated literals comprise the set of semantic units that are connected to objects and roles. A graphical representation of the complete LISA proposition is given in Figure 1. The two knowledge domains that are to be aligned in LISA are called the source analog and the target analog. The semantic units are shared among the two analogs; objects and roles of both analogs may be connected to the same set of semantic units.

2.2. Representing Ontologies in LISA

We assume that the ontologies are represented in the Web Ontology Language OWL. In encoding an ontology as a LISA analog, we build on the fact that LISA propositions are structurally similar to RDF triples, which have the form "subject – predicate – object" and, similar to first-order logic, consist of a function or relation symbol and two arguments. Thus, we first transform OWL ontologies to a set of RDF triples, in the form described in the OWL2 specification[1]. The top part of Table 2 shows two OWL class declarations with some annotations and a subclass axiom. The corresponding RDF triples are shown in the second part of the table. The bottom part of Table 2 shows the corresponding LISA analog definition. LISA keywords are highlighted in bold.

LISA Predicates. Each analog definition consists of the definition of predicates in the "Defpreds" section with their number of arguments and possibly an explicit definition of their roles (bottom part of Table 2). In our application, the set of predicates is fixed to represent OWL class, OWL object property, and OWL data property definitions, as well as OWL subclass axioms. Hence, we define four binary predicates, *Class*, *IsSubClassOf*, *ObjectProperty* and *DataProperty*. The roles for LISA predicates can either be defined implicitly or explicitly. An implicit definition occurs when only the arity (number of arguments) of each predicate is indicated, as for the *Class*, *ObjectProperty* and *DataProperty* predicates in the example in Table 2. When the roles are defined explicitly, they can be associated with semantic units, as in the *IsSubClassOf* example in Table 2.

[1] `http://www.w3.org/TR/owl2-mapping-to-rdf/`

Example	First-Order Logic	LISA
IsSubclassOf(SubClass/paper, SuperClass/publication)	Ground/closed atomic formula	Proposition
IsSubClassOf	Function symbol	Predicate
SubClass/paper	Substitution (variable binding)	Sub-Proposition
SuperClass/publication	Substitution (variable binding)	Sub-Proposition
SubClass	Variable	Role
Superclass	Variable	Role
paper	Ground term	Object
publication	Ground term	Object

Table 1: Representation of a first-order logic ground atomic formula in LISA terminology. The "A/a" symbol represents substitutation of variable A with ground term a.

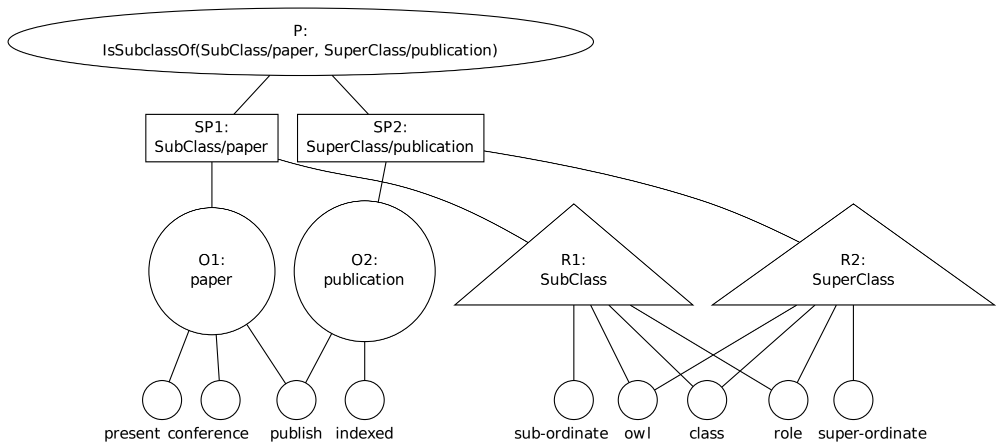

Figure 1: Representation of the statement *IsSubclassOf(SubClass/paper, SuperClass/publication)*. Propositions are shown as ovals (P), sub-propositions as rectangles (SP1, SP2), objects as large circle (O1, O2), roles as triangles (R1, R2), and semantic units as small circles (e.g. "conference"). The connections between units indicate the construction of the logic formulas and are also used for the connectionist-based algorithm to propagate action levels of units. They are undirected.

OWL
<owl:Class rdf:ID=”paper”>
<rdfs:label>paper</rdfs:label>
<rdfs:comment”>something that is presented at a conference and published. </rdfs:comment>
<rdfs:subClassOf rdf:resource=”publication”/>
</owl:Class>
<owl:Class rdf:ID=”publication”>
<rdfs:label>publication</rdfs:label>
<rdfs:comment”>something that is published and indexed. </rdfs:comment>
</owl:Class>
RDF triples
<paper> <rdf:Type> <owl:Class> .
<paper> <rdf:label> ”paper” .
<paper> <rdfs:comment> ”presented at a conference and published” .
<publication> <rdf:Type> <owl:Class> .
<publication> <rdf:label> ”paper” .
<publication> <rdfs:comment> ”something that is published and indexed” .
<paper> <rdfs:subClassOf> <publication> .
LISA
Analog ExampleAnalog1
Defpreds
Class 2 ;
IsSubClassOf [SubClass sub-ordinate role class OWL] [SuperClass super-ordinate role class OWL] ;
ObjectProperty 2 ;
DataProperty 2 ;
end ;
Defobjs
paper conference present publish ;
publication publish indexed ;
rdf:Type instance rdf classes ;
end ;
DefProps
P1 class (paper, rdf:Type) ;
P2 class (publication, rdf:Type) ;
P3 IsSubClassOf (paper, publication) ;
end ;

Table 2: Example OWL statements transformed via RDF triples to LISA format, corresponding to the example in Figure 1. See (Hummel and Holyoak, 1997) for further details about the LISA specification syntax.

The *Class*, *ObjectProperty* and *DataProperty* predicates are defined as binary predicates in the RDF triples, with the second argument always being the term "rdf:Type". While this may seem unnecessary, we have found that retaining this encoding actually improves LISA's performance as it provides additional structure to connect the different propositions in LISA's knowledge representation.

LISA Objects. Next, a LISA analog includes the definition of the objects with their semantic units in the "Defobjs" section (bottom part of Table 2). If an RDF triple is an OWL named class declaration, i.e. it has a *rdf:Type* predicate and an *owl:Class* object, its subject and object are represented as LISA objects. In other words, we represent OWL classes as LISA objects. Specifically, *rdf:Type* is a LISA object, and the RDF ID of the declared class becomes a LISA object. For example, the triple *<paper>* *<rdf:Type>* *<owl:Class>* causes "rdf:Type" and "paper" to be declared as LISA objects.

In a similar way, we generate LISA objects if the RDF triple has a *rdf:Type* predicate and an *owl:ObjectProperty* or *owl:DatatypeProperty* object. In other words, we also represent OWL properties as LISA objects.

Intuitively, when OWL classes and OWL properties are represented as LISA objects, we can then make statements in LISA about these objects, i.e. LISA propositions, which represent the statements in the OWL ontology about the classes and properties.

LISA Semantic Units. To generate the semantic units for each LISA object, i.e. literals that characterize each object, we parse labels and comments of named classes. In the example in Table 2, there are labels and a comment for the OWL classes *paper* and *publication*. While in this case the label is simply the same literal as the ID of the class, the comment provides additional characterization. We remove stopwords and stem the remaining terms of labels and comments. In this case, the terms "conference" and "present" are parsed from the comment for the *paper* class. These are defined as semantic units for the LISA object *paper* (Table 2). We add a fixed set of semantic units for each automatically generated LISA object. For example, we add the semantic units "instance", "classes", and "rdf" for the *rdf:Type* object because we believe these are characteristic of that object.

LISA Propositions. Finally, we use the declared LISA predicates and objects to construct propositions in LISA, found in the "DefProps" section in the bottom part of Table 2. Each proposition is labelled, e.g. "P1". In the example in Table 2, the proposition P1 expresses the fact that the LISA object *paper* is a class and is related to the *rdf:Type* LISA object.

When encoding a knowledge domain as a LISA analog, the default type of entity to be bound to a LISA role is a LISA object. Consider the following example of a class *paper* (P1) that has a subclass *journalPaper* (P2). The proposition P3 binds the LISA objects *journalPaper* and *paper* to the LISA roles of the *IsSubClassOf* predicate:

P1 Class $(paper, rdf : Type)$;
P2 Class $(publication, rdf : Type)$;
P3 IsSubclassOf $(paper, publication)$;

In an alternative encoding, a proposition is bound to a role in a similar way that ground atomic formulas can be bound to free variables in predicate logic. Given the same subclass relation between the *paper* class (P1) and the *journalPaper* class (P2), the alternative proposition P3' defined below encodes this relation explicitly by binding propositions P1 and P2 to the two roles of the *IsSubClassOf* predicate. We call this "relational encoding":

P3' IsSubClassOf$(P2, P1)$;

We emphasize that the way in which OWL ontologies are encoded in LISA analogs is not governed or constrained by any theoretical or logic-based principles. This is especially true for the semantic units, which are critical to the alignment process as they form the connection between the two LISA analogs. By parsing comments and labels, we have attempted to extract as much information from the ontology as possible. For "meta-level" objects, i.e. those that represent OWL constructs, we have relied on our intuition in creating characteristic and distinct sets of semantic units. For example, we believe that the literals "instance", "rdf", and "classes" characterize the *rdf:Type* object.

2.3. Connectionist Behaviour — Principles

Structurally, the connections between the units in LISA merely indicate the construction of the logic formulas as described above. However, the connections between the units are also important for the connectionist aspect of LISA: Units have an activation level, they can be activated ("fired") and the connections between units serve to "transmit" activation or inhibition to connected units (Section 2.4 below). The shared nature of the semantic units serves to transmit activation or inhibition between the two analogs.

While the terms "source analog" and "target analog" describe the direction in which the analogy is to be built, i.e. the direction of knowledge transfer, LISA uses the terms "driver analog" and "recipient analog" to indicate which of the two analogs begins the activation process. The driver analog is typically, but not necessarily, the source analog. Moreover, this direction can be changed in the course of the alignment process. In our application, the source analog is always the driver.

LISA operates by simulating activation over time of the units in the two analogs. Intuitively, this activation is LISA "paying attention" to those elements and is argued to mirror human attention mechanisms. The mappings of elements between the source and target analogs are determined by temporal co-activation patterns: When the activation pattern over time of an element of the target analog is similar to that of an element in the source analog, LISA creates a weighted mapping connection between the two elements. The weights increase if the activation levels over time of the source and target element are similar, and decrease otherwise. Intuitively, the idea is that if we pay attention to an element in the source analog, and this attention "brings to mind" an element in the target analog at the same time, then this represents an analogical mapping.

When a proposition is activated, it activates its sub-propositions. All units connected to the same sub-proposition are activated at the same time, and units of separate sub-proposition are activated at different times. Consider the timing diagram for the example in Figure 1 that is shown in Table 3. When the proposition P is activated, it activates its two sub-propositions, SP1 and SP2 asynchronously. When a sub-proposition is activated, it in turn activates the object and role units to which it is connected. For example, the activations of O1 and R1 occurs at the same time points as that of SP1. The object and role units in turn activate the semantic units they are connected to. Because semantic units may be connected to multiple object or role units, and because object and role units in turn may be connected to multiple sub-propositions, a complex temporal pattern of activation is generated on the semantic units. The activation then propagates to the recipient analog.

Repeat until all P units have been activated:

1. Select a set of P units of the driver analog ("phase set P_S"). Set the input to any SP unit that is connected to a P unit in the phase set to 1.
2. Repeatedly update the state of the network in discrete time units $t = 1 \ldots 220 \times |\mathrm{SP} \in P_S|$. For each step do:
   (a) Update modes of all P units in the recipient
   (b) Update inputs of all P units in P_S and their connected SP, role and object units
   (c) Update the global inhibitor
   (d) Update the inputs to all semantic units
   (e) Update the input to units in recipient
   (f) Update the activation of all units
   (g) Create mapping connections
3. Update mapping weights

Table 4: LISA Algorithm from (Hummel and Holyoak, 1997) (P = Proposition, SP = Sub-proposition)

2.4. Connectionist Behaviour — Algorithm

The input to the algorithm are two analogs that encode the relevant domain knowledge and that share some semantic units, as described above. The output of the algorithm is a set of weighted mapping connections between units of the source and target analog (possibly connecting many target units to each source unit and vice versa) with the weight normalized to the interval $[0, 1]$ for each mapping connection. Table 4 shows the main steps of the LISA algorithm.

Step 1. Any number of propositions can be selected into the phase set. In our application, we typically select one, two, or three propositions in different orders into the phase set. In the experimental study described in Section 3, this is controlled by the experimental design factors "Ordering" and "Updates", where the latter controls how many propositions are activated (i.e. are in the phase set) before mapping weights are updated in step 3.

		Time →						
P	IsSubclassOf(SubClass/paper, SuperClass/publication)	x	x	x	x	x	x	x
SP1	SubClass/paper	x		x		x		x
O1	paper	x		x		x		x
R1	Subclass	x		x		x		x
SP2	Superclass/publication		x		x		x	
O2	publication		x		x		x	
R2	Superclass		x		x		x	
	publish	x	x	x	x	x	x	x
	conference	x		x		x		x
	present	x		x		x		x
	indexed		x		x	x		x
	...	...						
	owl	x	x	x	x	x	x	x

Table 3: Activation of units in the example in Figure 1 against time.

Step 2a. P units operate in either parent, child or neutral mode. Initially, P units in the driver analog are in neutral mode, except those selected into the phase set in step 1, which are in parent mode. In parent mode, P units pass activation input downwards to their SP units. In neutral mode, P units pass activation input both upwards and downwards. In child mode, P units pass activation input upwards to the SP units they are part of (recall that propositions can take the place of objects in subpropositions). P units in the receiver analog update their mode on the basis of their inputs from SP units above ($SP^{\uparrow}$) and below ($SP^{\downarrow}$), and also on the basis of their inputs from P units in the driver that are either in parent mode (P^{parent}) or child mode (P^{child}) to which they may be connected via mapping connections:

$$m = \begin{cases} \text{Parent} & \text{if } SP^{\uparrow} - SP^{\downarrow} \\ & \quad +P^{parent} - P^{child} > \theta \\ \text{Child} & \text{if } SP^{\uparrow} - SP^{\downarrow} \\ & \quad +P^{parent} - P^{child} < -\theta \\ \text{Neutral} & \text{otherwise} \end{cases}$$

Step 2b. Each SP unit consists of an excitor with activation e and an inhibitor with activation I. In the driver, SP excitors receive excitatory input from P units above (p), inhibitory input from other driver analog SP excitors ($\sum_j e_j$) with a sensitivity s that decays with each simulated time slice, and inhibitory input from their own inhibitor (I). The net input to an SP excitor is

$$n = 1 + p - \frac{s \sum_j e_j}{1 + \text{NSP}} - I$$

where NSP is the number of SPs in the driver with activation > 0.2.

An SP inhibitor receives input only from the corresponding excitor and changes its activation according to

$$\Delta I = \begin{cases} \gamma^s & I \leq \Theta^L, e > \Theta^I \\ \gamma^f & I > \Theta^L, e > \Theta^I \\ -\delta^s & I \geq \Theta^U, e \leq \Theta^I \\ -\gamma^f & I < \Theta^U, e \leq \Theta^I \end{cases}$$

where γ are growth rates, Θ are lower and upper thresholds, and δ is a decay rate.

P units in parent mode receive input from the excitors of the SP units below with net input $n = \sum_j e_j$, where e_j is the activation of the excitors of SP unit j. P units in child mode, role or object units receive input from the excitor and the inhibitor of the SP above with net input $n = \sum_j (e_j - I_j)$, where I_j is the activation of the inhibitor of SP unit j.

Step 2c. The global inhibitor is set to $\Gamma = 10$ if all SP excitors in the driver analog have an activation below the threshold Θ^G.

Step 2d. Semantic units receive weighted input from role and object units in both the driver and receiver analog, with the net input computed as $\sum_j a_j w_{ij}$, where w_{ij} are the connection weights of semantic unit i to role or object unit j, and a_j serves to normalize activation to between 0 and 1. Input from the receiver analog occurs only after each SP in the driver has been activated at least once.

Step 2e. Units in the recipient analog receive five inputs:

- within-proposition excitatory input P,
- within-class inhibitory input C (i.e. SP-to-SP, object-to-object, etc.),
- out-of-proposition inhibitory input O (i.e. P units inhibit SP units of other P units, and SP units inhibit role and object units of other SP units),
- both excitatory and inhibitory input M via the cross-analog mapping relationships, and
- inhibitory input from the global inhibitor $\pi\Gamma$ (Step 2c).

The net input n is computed as

$$n = P - C - \pi O + M - \pi\Gamma$$

For P units, the within proposition excitatory input P is a function of the activation of the excitors of the SP units above and below the P units:

$$P = (1-\pi)\sum_j e_j + \pi \sum_k e_k$$

When P is in parent mode, $\pi = 0$, otherwise $\pi = 1$, j are SP units below and k are SP units above.

The within class input C for a P unit i is defined as

$$C_i = \frac{\sum_{j \neq i} a_j m(i,j)}{1+n}$$

where a_j is the activation of other P units, n is the number of P units that are activated above a certain threshold Θ^I; $m(i,j) = 1$ if i is in neutral mode or if i and j are in the same mode, and is 0 otherwise.

The out-of-proposition inhibition is defined as the sum of the activation a_j of P units j that are neither below or above it:

$$O_i = \sum_j a_j$$

Finally, the input M_i from mapping relationships for a P unit i is defined as

$$M_i = \sum_j m(i,j) a_j (3w_{ij} - max(w_{i.}) - max(w_{.j}))$$

where a_j is the activation of P unit j in the target analog, w_{ij} is the mapping weight between node i and node j in the target analog and $w_{i.}$, and $w_{.j}$ are means over j and i respectively.

For SP units, the terms for the net input are defined as above for P units, except for the within-proposition input P, which is defined as

$$P = p + r + o + c$$

where p is the activation of the P unit above if that unit is in parent mode and 0 otherwise, r is the activation of the predicate unit below, o is the activation of the object unit below (if there is one, otherwise it is 0), c is the activation of any P unit serving as child (if there is one, otherwise it is 0).

For the role and object units, the terms for the net input are defined as above for P units, except for the within-proposition input P, which is defined as:

$$P_i = \frac{\sum_{j=1}^{n} a_j w_{ij}}{1+n} + \frac{\sum_{k=1}^{m} a_k}{1+m}$$

Here, j are the semantic units to which i is connected and a_j their activation; k are the SP units above this role or object unit and a_k their activation; w_{ij} are the connection weights.

Step 2f. At the end of each iteration and after computation of all excitatory and inhibitory input, all units update their activation according to $\Delta a = \gamma n(1-a) - \delta a$ where γ is a growth rate and δ is a decay rate; n is the net input (excitatory minus inhibitory).

Step 2g. When the activation of a unit in the recipient analog first exceeds 0.5, mapping connections are created between it and units of the same type in the driver analog. These connections and their weights are important for the subsequent iteration.

Step 3. Each mapping weight w_{ij} is updated, taking into account competing mapping hypotheses and ensuring a 1:1 mapping constraint.

2.5. Connectionist Behavior — Worked Example

We present a brief example of the algorithm using the single analog shown in Figure 1 and Table 2. Given the iterative nature of the algorithm and its complexity, we can only show a few steps.

In step 1, proposition *P1* is selected into the phase set and activated. This sets the inputs to *SP1* and *SP2* to 1 (step 1). The following steps are repeated $220 \times 2 = 440$ times.

In step 2a, the P units in a recipient analog are not yet connected to any P units in the driver, so P^{parent} and P^{child} are 0, as is the input from SP units above or below $(SP^{\uparrow}, SP^{\downarrow})$.

In step 2b, the excitor for *SP1* receives excitatory input receive excitatory $p = 1$ from its P unit. The other excitors are not yet active in this iteration, so that $e_j = 0$, and neither is the inhibitor active yet, so that $I = 0$. Thus, $n = 2$ for the excitor of *SP1* in this iteration. Note that the excitor is not actually updated until the end of the iteration (step 2f), so $e_j = 0$. The change to the inhibitor ΔI is set using the growth and decay rates according to the formula specified in step 2b. Since $I = 0 < \Theta^L$ and $e_j = 0 > \Theta^I$ in this iteration (assuming typical thresholds Θ^I and Θ^L), the inhibitor is set to grow according to a growth rate γ. Also as part of step 2b, the *P1* unit, which is in parent mode, receives input from the excitors of *SP1* and *SP2*, but this net input is still $n = 0$ as the excitors are not actually updated until the end of the iteration (step 2f). There are no P units in child mode. However, role and object units connected to *SP1* and *SP2* receive net input $n = 0$ in this iteration. Again, this is because neither the excitor nor inhibitor actually update until the next iteration.

In step 2c, the global inhibitor is not activated, as the excitors in the driver have only just become active.

In step 2d, the semantic units receive activation from the role and object units. The initial weights are set to 1. There is not yet any input to the semantic units from the receiver analog.

In step 2e, the units in a receiver analog are activated according to the description above. The P units in the receiver do not yet receive any within-proposition activation, as the excitors of their SP units are not yet active. They do not yet receive any within-class input or out-of-proposition inhibition activation either, as the other P units in the receiver are not yet activated. There is also no input from mapping relationships, as there are no mapping relationships yet. The same is true for the SP, role and object units in the receiver.

In step 2f, all units update their input according to the computed net change and the formula presented above. For example, the excitor for *SP1* with a computed net change of $n = 2$ and a current activation of $e = 0$ will, assuming a growth and decay rate $\gamma = \delta = 0.5$, be changed by $\delta e = 0.5 \times 2(1 - 0) - 0.5 \times 0 = 1$.

In step 2g, mapping connections are created between units of the same type that are active at > 0.5 at this time in both the driver and the recipient.

3. Experiment, Results, and Comparison

We evaluated LISA's performance on ontology alignment tasks using standard benchmark tests. We first describe the benchmark ontology alignment tests. Then we present our results and a comparison with existing approaches. Our evaluations used the publicly available version of LISA[2].

3.1. Benchmark Dataset

To evaluate LISA's performance, we used the set of alignment tests available from the Ontology Alignment Evaluation Initiative (OAEI) (Aguirre et al., 2012; Shvaiko and Euzenat, 2013). We used the "benchmark" test set, a systematic benchmark series that is based on a reference ontology in the domain of academic bibliography. It has 33 named classes, 24 object properties, 40 data properties, 56 named individuals and 20 anonymous individuals. This test set is intended to be stable, with results published annually since 2004. The tests are grouped into three main categories:

1. *Simple tests (1xx):* The reference ontology is compared with itself, its representation in OWL-Lite, or an irrelevant ontology. To determine the correct operation of LISA, we have only implemented the first of these tests, the comparison of an ontology with itself.
2. *Systematic tests (2xx):* Various modified versions are compared with the reference ontology, see Table 8. There are 93 tests in this set.
3. *Real ontologies (3xx):* The reference ontology is compared with other bibliography domain ontologies that have been independently developed.

[2] `http://internal.psychology.illinois.edu/~jehummel/models.php`

While the tests in the 300-series are perhaps more realistic than the synthetic benchmarks in the two other categories, the reference alignments for these tests are known to be flawed[3]. Thus, our evaluation instead focuses on the 200-series of tests. In these tests, the reference ontology is systematically varied by suppressing the names and/or comments, by suppressing, flattening or expanding the specialization hierarchy, by suppressing the properties of concepts, or by various combinations of these changes (Table 8). Figures B.5 and B.6 in the appendix show, as an example, the two ontologies to be matched for OAEI benchmark test 258. Given the space limitations and the complexity of the ontologies, only the class hiearchies are shown.

3.2. Experimental Design

LISA, and our encoding of OWL ontologies in LISA, is configurable. We investigated different configurations in order to better understand their influence on the alignment results and to find an optimal configuration for comparison to state-of-the-art alignment systems. We identified and varied the levels of four experimental design factors.

1. As described in Section 2.2, LISA allows the binding of either objects or propositions to roles. We refer to the first encoding as "non-relational" and to the second "relational".
2. While the activation order of propositions in the original LISA work (Hummel and Holyoak, 1997) was optimized manually, this is not feasible in the ontology alignment context where the analogs are much larger. Instead, we have defined and experimented with the following proposition activation orders:
   - *Ordered:* All propositions are activated in the order in which they appear in the ontology. When propositions are activated multiple times, the same sequence is repeated.
   - *OrderedCSC:* First, propositions representing classes are activated, then propositions representing subclass relationships, then proposition representing object and data properties. When propositions are activated multiple times, the same activation sequence is repeated.
   - *Grouped:* Similar to OrderedCSC, propositions are activated in groups, first those representing classes, then those representing sub-class relationships, then those representing object and data properties. However, when propositions are activated multiple times, the set of propsitions representing classes is activated multiple times, followed by the set of propositions representing subclasses activated multiple times, followed by the set of propositions representing object and data properties, activated multiple times.
   - *Random:* The propositions are activated in random order.
3. LISA can be configured to activate propositions more than once, representing repeated "attention" being paid to propositions. As described above, this has different effects depending on the chosen ordering of propositions. We investigate activating propositions once or twice.
4. LISA normally updates mapping connections and their weights after the activation of each proposition (step 3 of the algorithm in Table 4). We also investigate activating two or three propositions before mappings are updated.

Table 5 shows a summary of the four experimental design factors. Their combination results in 48 experimental conditions. For each of these conditions, we applied LISA to all tests in the reference alignment test set.

To evaluate the performance of LISA, the obtained alignments are compared to a set of reference alignments that are provided as part of the OAEI benchmark set. We use the the F-score measure, which combines the precision and recall metrics, to evaluate our ontology alignment results. Figures 2 and 3 show graphical representations of the mean and median F-score across all 94 OAEI benchmark tests under different experimental conditions. The full results can be found in tables A.9 and A.10 in the appendix. The figures clearly show that the median values are generally higher than the means, and also illustrate the complex interaction patterns between the experimental design factors.

3.3. Data Analysis

To identify which of the experimental design factors had a significant effect on the F-scores, we per-

[3] `http://oaei.ontologymatching.org/2012/benchmarks/`

Design Factor	Description	Levels (Base line in *emphasis*)
Encoding	Propositions or objects as role fillers (Sec. 2.2)	*Objects ("Non-Relational encoding")*, Propositions ("Relational encoding")
Ordering	Order of proposition activation (Sec. 2.4, Step 1)	*Ordered*, OrderedCSC, Grouped, Random
NumProps	Number of times each proposition is activated (Sec. 2.4, Step 1)	*1*, 2
Updates	Number of propositions to activate before mappings are updated (Sec. 2.4, Step 3)	*1*, 2, 3

Table 5: Experimental Design Factors

	Estimate	Std. Error	t value	Pr(>|t|)
(Baseline)	0.4375	0.0111	39.35	0.0000
Encoding.Rel	0.0149	0.0079	1.90	0.0574
Ordering.Grouped	0.0679	0.0111	6.11	0.0000
Ordering.OrderedCSC	0.0836	0.0111	7.52	0.0000
Ordering.Random	0.0452	0.0111	4.07	0.0000
Updates.2	0.0177	0.0096	1.84	0.0657
Updates.3	0.0040	0.0096	0.42	0.6778
NumProps.2	0.0264	0.0079	3.36	0.0008

Table 6: ANOVA Results: Effects of Experimental Design Factors on F-Score and Statistical Significance (Main effects only, no interaction effects)

formed an analysis of variance (ANOVA), limited to the main effects of the four design factors without including interaction effects. The results are shown in Table 6 with effects compared to the base line configuration that is highlighted in Table 5. First, the mean F-score across all tests in the baseline configuration is 0.4375. Next, we examine the effect of the experimental design factors on the F-score:

Encoding. The encoding of the ontology in LISA (design factor "Encoding" in Tables 5 and 6) did not have a significant effect on the F-score ($p = .0574$). A slightly higher mean F value (.4524) is achieved by moving to relational encoding. As expected, LISA is primarily a structural matcher and can exploit the increased structure in the encoding that provides connection between different propositions, rather than only between objects.

Ordering. Our results indicate that the activation order for propositions (design factor "Ordering" in Tables 5 and 6) has the largest effect on the F-scores. The mean F-score improves with any of the other activation orders, and the increases are statistically significant. Specifically, activating the propositions in groups improves the mean F-score by .0679 against the baseline, and the activation order "OrderdCSC" improves the F-score by .0836. The ordering is important due to the iterative nature of the algorithm: Mapping connections are created and their weights are updated (steps 2g and 3) after each proposition is activated (or after two or three activated propositions in some conditions). Hence, the activation levels created by earlier activations have an effect on the updated connections and weights. Consequently, a different activation order is likely to result in different mapping connections and weights.

Updates. The number of propositions h that are activated before mappings are updated (step 3 in Table 4, design factor "Updates" in Tables 5 and 6) has no significant effect on the F-scores. For $h = 2$, there is an increase of approx. 2%. Interestingly, for the relational encoding, there is a significant positive effect (mean $F = 0.53745$) for $h = 3$, while there is a significant negative effect (mean $F = .48528$) for $h = 3$ and non-relational encoding.

Number of Propositions. Repeating the activation of propositions (design factor "NumProps" in Tables 5 and 6) also has a small but significant effect on the F-scores, increasing the F-score by .0264

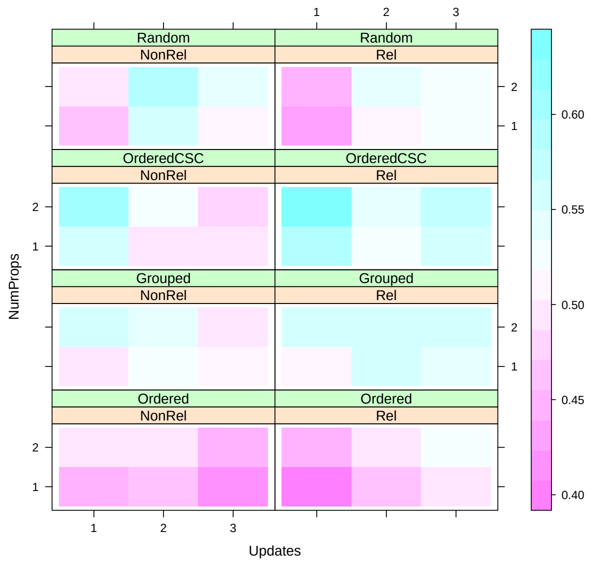

Figure 2: F-scores under different experimental conditions (mean across all OAEI tests)

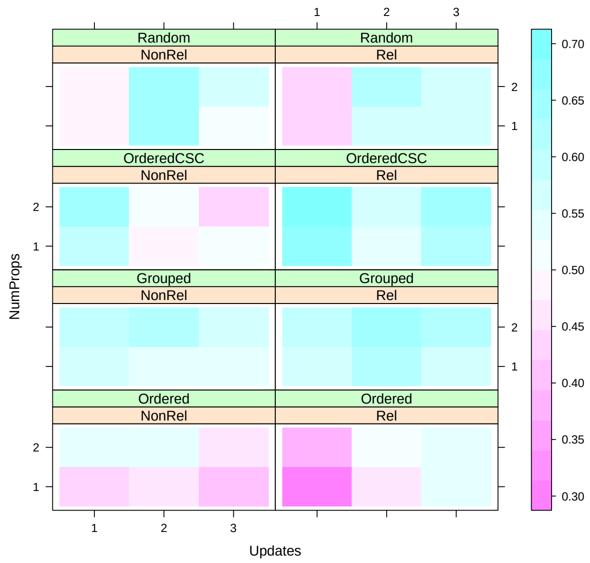

Figure 3: F-scores under different experimental conditions (median across all OAEI tests)

against the baseline. This is expected, as the algorithm can update the mapping connections and their weights again after the final proposition in the first sequence has been activated, in a sense allowing LISA to "revisit" and "refine" earlier mappings in light of propositions activated later.

In addition to these main effects, there are a variety of interaction effects, such that changes to the F-score depend on multiple experimental factors. The LISA configuration with the highest mean F-score across all 94 OAEI benchmark tests was relational encoding, ordered by class–subclass groups, with a single proposition activated between mapping updates, and activating propositions twice. The mean F-score for this condition was 0.6293. This was also the configuration with the highest median F-score across all OAEI tests; the median F-score for this condition was 0.6866. Using the optimal LISA configuration for each OAEI benchmark test, the mean F-score across all OAEI tests is 0.7147 and the median F-score across all OAEI tests is 0.7788.

3.4. Comparison to Other Systems

Table 7 shows the mean and median F-scores of all reported systems in the 2012 OAEI competition[4]. Based on this data, LISA in a single optimal configuration across all tests ranked fourth by mean and sixth by median F-score. When using an optimal configuration for each OAEI test (labelled "LISA (opt.)" in Tab. 7), LISA again ranks fourth by mean F-score (with a very small difference behind AROMA) and also fourth by median F-score, with only a small difference behind AROMA.

Test cases such as the OAEI benchmarks are intended to be representative of a range of different situations, e.g. differing by knowledge domain or by the type of ontology features. Thus, one can interpret the benchmark tests as samples from a possible set of benchmarks. With this interpretation, one can ask whether the observed F-score differences between LISA and other systems on this particular set of benchmarks indicate a statistically significant difference in the population of alignment problems. To answer this question, we compare LISA to other systems using a pairwise comparison test. For this, we use the single best LISA configuration (relational encoding, ordering by class–subclass, propositions activated twice, and mapping

[4]The OAEI results compute the mean F-score by first computing the arithmetic means of precision and recall across all benchmark tests, and then computing the harmonic mean. However, this is not the mean of the F-scores of different tests, but the F-score of the mean precision and recall and has a tendency to artificially inflate the results. In this paper, we first compute the harmonic mean of precision and recall for individual tests, and then take the arithmetic mean of those.

System	Mean F	Median F	Sig. Diff.
MapSSS	0.7784	0.8732	*
YAM++	0.7682	0.8529	*
AROMA	0.7169	0.7922	*
LISA (opt.)	0.7147	0.7788	*
LISA	*0.6294*	*0.6866*	
WeSeE	0.6069	0.7461	
AUTOMSv2	0.6068	0.7414	
GOMMA	0.5842	0.6690	*
Hertuda	0.5826	0.7100	*
HotMatch	0.5706	0.6604	*
Optima	0.5520	0.6073	*
MaasMatch	0.5488	0.6051	*
Wmatch	0.5383	0.6198	*
ServOMap	0.5335	0.5663	*
LogMapLt	0.5065	0.5567	*
LogMap	0.4883	0.5174	*
MEDLEY	0.4802	0.5244	*
ASE	0.4569	0.5075	*
edna	0.4110	0.4000	*
ServOMapLt	0.3581	0.3333	*

Table 7: Mean and Median F-scores of systems in the 2012 OAEI competition (across 94 benchmark tests), sorted by mean F with significant differences (paired t-test, $\alpha < 0.05$) to LISA score indicated. "LISA" represents LISA in single optimal configuration; "LISA (opt.)" represents LISA with test-specific optimal configurations.

updates after each activated proposition). Performing this comparison to all other systems shows that the differences in F-score are significant only for some systems (Table 7, column "Sig. Diff."). There are significant differences between LISA's performance and that of the group of top three other approaches. Further, there are significant differences between LISA's performance and the bottom group of 13 other approaches. Thus, while there may be differences for this particular set of benchmark tests, these sample differences do not necessarily reflect real performance differences across different but similar benchmark tests, e.g. with a different knowledge domain. Not indicated in Table 7, when LISA is optimized for each test ("LISA (opt.)"), there are no significant differences to the AROMA system, indicating that LISA in the test-optimal configuration occupies third place in the ranking.

3.5. Effects of Test Difficulty

Many of the tests in the OAEI benchmark suite come in different levels of difficulty as the benchmark tests are synthetically created by altering the source ontology. For example, the target ontology for test 201 is created by omitting names from the common baseline ontology. For many tests, the changes are applied for only a fraction of the ontology, rather than the entire ontology. For example, test 201-2 omits names from 20% of the ontology, 202-4 omits names from 40% of the ontology, etc., resulting in increasingly difficult alignment tests. The OAEI benchmark provides tests 201-202, 248-254, and 257-262 in these different levels of difficulties. As indicaed in Table 8, these tests deal primarily with the suppression of names and comments, but also include variations of the hierarchy and the object and data properties (tests 248-254, 257-262).

Figure 4 shows a plot of the F-values for the optimal LISA configuration for tests that have different levels of difficulties. As expected, the performance decreases as more names and comments are removed, although there is considerable variation in the results. The different tests are closer in performance when more names and comments are present, and performance increasingly diverges as such information is removed. We conclude that the information about hierarchy, instances, and classes that is manipulated by these tests is increasingly important as the information that is contained in names and comments is removed. While LISA is primarily a structural alignment algorithm, as described in Sec. 2, the names and comments are critical for our encoding, as they are parsed to form the semantic units that connect the two analogs in LISA (Sec. 2.2).

3.6. Effects of Ontology Features

Table 8 shows how the synthetic benchmarks manipulate different ontology features. It also shows the F-score achieved by LISA in its optimal configuration (propositions ordered by classes–sub-classes, propositions activated twice, mapping updated after each activated propositions) for both types of encodings ("Relational', "NonRelational"). The ontology features manipulated by each test affect the performance of relational and non-relational encoding in different ways.

Names and Comments. A closer look at the benchmark tests reveals the importance of names and comments in the alignment task and its effect on the performance of the two types of ontology encodings. For example, in test 101 and tests 221-247

	Concepts				Encoding	
Test #	Name	Comment	Hierarchy	Properties	Non-Relationa	Relational
101	-	-	-	-	92	96
201	R	-	-	-	81	87
202	R	S	-	-	27	24
221	-	-	S	-	85	89
222	-	-	F	-	68	94
223	-	-	E	-	90	93
224	-	-	-	-	91	96
225	-	-	-	R	84	86
228	-	-	-	S	95	98
232	-	-	S	-	89	85
233	-	-	S	S	86	98
236	-	-	-	S	71	82
237	-	-	F	-	91	86
238	-	-	E	-	87	92
239	-	-	F	S	84	98
240	-	-	E	S	73	95
241	-	-	S	S	84	98
246	-	-	F	S	97	98
247	-	-	E	S	75	84
248	S	S	S	-	30	24
249	S	S	-	-	10	0
250	S	S	-	S	71	57
251	S	S	F	-	25	22
252	S	S	E	-	34	27
253	S	S	S	-	10	0
254	S	S	S	S	65	50
257	S	S	-	S	11	6
258	S	S	F	-	8	2
259	S	S	E	-	12	2
260	S	S	F	S	50	47
261	S	S	E	S	60	50
262	S	S	S	S	22	6
265	S	S	F	S	17	6
266	S	S	E	S	17	0

Note: S refers to suppressed, R to replaced, F to flattened, and E to expanded

Table 8: OAEI benchmark tests and LISA F-score values between relational and non-relational encodings for the optimal LISA configuration. Tests with partially modified target ontology not listed for space reasons.

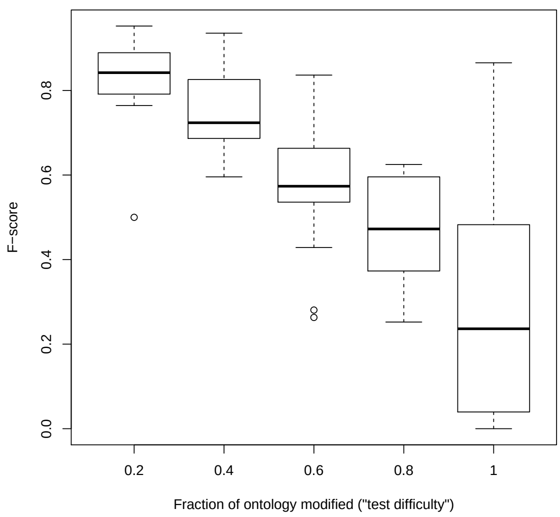

Figure 4: Boxplot of F-values for different levels of test difficulty for the optimal LISA configuration

where entity names and comments were available, the F-scores were significantly higher for the relational encoding. For tests 248-266, where names and comments were suppressed, the non-relational encoding performed better.

While LISA is primarily a structural matcher, one critical element in its architecture is the connection of the two analogs via shared semantic units. As described in Section 2, these are parsed from labels and comments in natural language. When these are suppressed there are only a few shared semantic units left to connect the two analogs, those defined with reference to OWL terminology, e.g. semantic units such as "OWL", "class", "rdf", "subclass", etc. The diminished performance in conditions without comments and other annotations is therefore not surprising.

Hierarchy. Suppressing, flattening, or expanding the ontology subclass hierarchy also affected the alignment results. First, suppressing the hierarchy added an additional level of difficulty to the task of alignment as LISA is unable to exploit ontology structure and must increasingly rely on the shared semantic units. However, the F-scores for both relational and non-relational encodings remained high (relative to the equivalent tests with hierarchy information present) in tests 221, 232, 233, 241, 248, 253, 254, and 262. Second, flattening the ontology hierarchy leads to a general decrease in the F-scores, seen particulary in tests 222, 237, 239, and 246, and even more so in tests 251, 258, 260, and 265. In the latter case, LISA has neither textual information nor structural information available. Third, as expected, expanding the hierarchy of the ontologies leads to good results, seen in tests 223, 238, 240, and 247. These conditions provide additional structural information that can be exploited.

Properties. Surprisingly, suppressing properties from the ontologies results in a better F-scores for both relational and non-relational encodings for some tests. For example, comparing test 222 with test 239 shows that LISA achieved better results when the properties were suppressed. However, LISA performed better when all properties were suppressed rather than only removing restrictions on classes: compare test 225, where properties restrictions on classes have been discarded, with test 228, where properties and relations between objects have been completely suppressed.

This seemingly surprising result may be explained by the fact that, while properties add important structural information to the problem, they must also be matched, and this adds a level of difficulty to the problem in the form of a significantly increased set of concepts that must be aligned. This is especially true for data properties as these add little structure to the problem. In the absence of naming or other information, two data properties on the same class are essentially indistinguishable. In contrast, object properties connect classes and thus add structure that LISA can exploit.

4. Discussion and Conclusion

This paper has investigated ontology alignment from the perspective of analogical reasoning. Specifically, we chose LISA as our first cognitive model of analogy to be explored in this line of research.

The paper makes three specific contributions. First, from the cognitive science perspective, we adapt and apply the LISA system to a novel context. This extension and the systematic exploration of LISA performance under varying conditions and on different tests can be used to further help refine this computational model of analogical reasoning. Second, from the perspective of semantic web research, we have demonstrated that analogical reasoning can provide a good source of algorithms and heuristics to tackle the difficult problem of ontology alignment. Finally, our research bridges the gap between the disciplines of cognitive science and semantic web research; it is a step towards the vision of a cognitive semantic web (Raubal and Adams, 2010; Gentner et al., 2012). To our knowledge, it is the first application of a cognitive model to the ontology alignment problem. We believe that our proposed approach to combine cognitive science and the semantic web shows promise. The results we presented are approximately equivalent or surpass those of the state-of-the art alignment systems. However, other cognitive models of analogy remain to be explored and more work remains to be done, both with LISA and with other cognitive computational models. Specifically, we have identified the following challenges for future research.

First, we applied LISA and configured its encodings, mapping updates, proposition activation order, and the number of time propositions are activated. We observed that LISA's performance depends strongly on the order in which propositions are activated. This is especially true when propositions are activated multiple times. The ordering is important due to the iterative nature of the algorithm as the activation levels created by earlier activations have an effect on the mapping connections and weights used in later activation updates. Prior research on LISA suggests that when multiple propositions are activated before the next mapping connection and weight update, they should be chosen in ways that create "textual coherence", e.g. be related to the same higher-order relation or share a number of arguments and predicates (Hummel and Holyoak, 1997). Thus, detecting such related propositions to guide the algorithm is a key challenge to achieving good performance. While we have explored some proposition activation orders in this research, many others could be designed that may lead to improved results.

Second, as we have noted earlier, the encoding of ontologies in LISA admits many design decisions and is not governed or constrained by any theoretical or logic-based principles. We have chosen an intuitive route via an RDF representation of OWL ontologies and a direct mapping to LISA knowledge representation elements. Other encodings are possible. For example, many of the OWL constructs that are now encoded as semantic units may be represented as LISA objects in their own right. The latter approach may strengthen the structure of the problem representation, yet at the same time reduce the number of semantic units by which activation progresses from driver to recipient. Another decision is the extent to which the structure of OWL should be represented. For example, one could imagine to represent the entire OWL vocabulary as LISA predicates and construct propositions that directly reflect the OWL ontology. On the other hand, one can limit the use of OWL vocabulary as we have done here and instead focus on the content of the ontology. The first approach will likely yield "deeper" structure in LISA at the expense of being specific to OWL ontologies and perhaps overemphasizing structural information. The latter approach may yield a "flatter" structure in LISA, but may afford LISA more flexibility in finding matches. The effects of such different encodings remains to be explored in further systematic experiments.

Third, similar to all other state-of-the-art systems, the performance of LISA was significantly lower when information from labels and comments was expressed in a different language. Thus, one important question concerns the translation of labels, comments, and annotations of concepts. While LISA is primarily a structure-based algorithm, the key to its performance is the set of shared semantic units. When ontologies are presented in different languages, semantic units are unlikely to be shared, severely limiting the performance of LISA. This would suggest that ontologies should be pre-processed by translating terms to a common language, using information and services external to the ontologies. This is done by all other systems that participated in the 2012 OAEI contest. On the other hand, this external information should strictly not be necessary as LISA itself is presented as a way to "translate" concepts and may therefore not reflect cognitive principles (Hummel and Holyoak, 1997, 2003).

The final challenge concerns the reasoning support that is afforded by ontologies. As OWL ontologies admit inferences over the subsumption hierarchy, it is possible to also use inferred subsumption relationships in the ontology alignment algorithm, in addition to those explicitly axiomatized in the ontology. This is exploited by some of the systems that participated in the OAEI contest and may be used to improve the alignment performance. However, as with the translation issue, the important question from the cognitive perspective is whether such formal reasoning still represents human cognitive processes.

In conclusion, our research has contributed to a theoretical foundation of ontology alignment in the psychology of analogical reasoning, and opened up possibilities for future work in the area. This foundation can provide guidance for the field of ontology alignment in that results from the ongoing work in cognition may be transferred to improve ontology alignment performance.

Aguirre, J., Eckert, K., Euzenat, J., Ferrara, A., van Hage, W., Hollink, L., Meilicke, C., Nikolov, A., Ritze, D., Scharffe, F., Shvaiko, P., Svab-Zamazal, O., Trojahn, C., Jomenez-Ruiz, E., Ceuenca Grau, B., Zapilko, B., 2012. Results of the ontology alignment evaluation initiative 2012. In: Proceedings of the 7th International Workshop on Ontology Matching, Boston, MA.

Aufaure, M., Le Grand, B., Soto, M., Bennacer, N., 2006. Metadata-and ontology-based semantic web mining. In: Web Semantics and Ontology. Idea Group Publishing, pp. 259–296.

Berners-Lee, T., Hendler, J., Lassila, O., 2001. The semantic web. Scientific American 284 (5), 34–43.

Cruz, I. F., Antonelli, F. P., Stroe, C., 2009. Agreementmaker: Efficient matching for large real-world schemas and ontologies. Proceedings of the VLDB Endowment 2 (2), 1586–1589.

Evermann, J., 2008a. An exploratory study of database integration processes. IEEE Transactions on Knowledge and Data Engineering 20 (1), 99–115.

Evermann, J., 2008b. Theories of meaning in schema matching: A review. Journal of Database Management 19 (3), 55–83.

Evermann, J., 2009. Theories of meaning in schema matching. Information Systems 34 (1), 28–44.

Evermann, J., 2010. Contextual factors in database integration — a Delphi study. In: Proceedings of the 29th International Conference on Conceptual Modeling (ER 2010), Vancouver, BC.

Evermann, J., 2012. Applying cognitive principles of similarity to data integration–the case of SIAM. In: Proceedings of the America's Conference on Information Systems (AMCIS), Seattle, WA.

Falkenhainer, B., Forbus, K. D., Gentner, D., 1989. The structure-mapping engine: Algorithm and examples. Artificial Intelligence 41 (1), 1 – 63.

Forbus, K., Gentner, D., Markman, A., Ferguson, R., 1998. Analogy just looks like high level perception – Why a domain-general approach to analogical mapping is right. Journal of Experimental & Theoretical Artificial Intelligence 10 (2), 231–257.

Forbus, K. D., 2001. Exploring analogy in the large. In: Analogy – Perspectives from Cognitive Science. Cambridge, MA: MIT Press, pp. 23–58.

French, R. M., 2002. The computational modeling of analogy-making. Trends in Cognitive Sciences 6 (5), 200 – 205.

Gentner, D., 1983. Structure-mapping: A theoretical framework for analogy. Cognitive science 7 (2), 155–170.

Gentner, D., 2003. Why We're So Smart. In: Language in Mind – Advances in the Study of Language and Thought. MIT Press, Cambridge, MA, pp. 195–235.

Gentner, D., 2010. Psychology in cognitive science. Topics in Cognitive Science 2 (3), 328–344.

Gentner, D., Forbus, K. D., 2011. Computational models of analogy. Wiley Interdisciplinary Reviews – Cognitive

Science 2 (3), 266–276.
Gentner, D., van Harmelen, F., Hitzler, P., Janowicz, K., Khnberger, K., 2012. Cognitive approaches for the semantic web. Dagstuhl Reports 2 (5), 93–116.
Gick, M. L., Holyoak, K. J., 1980. Analogical problem solving. Cognitive Psychology 12 (3), 306 – 355.
Goldstone, R. L., Medin, D. L., 1994. Time course of comparison. Journal of Experimental Psychology: Learning, Memory, and Cognition 20 (1), 29.
Gruber, T. R., 1993. A translation approach to portable ontology specifications. Knowledge Acquisition 5 (2), 199–220.
Gust, H., Krumnack, U., Kühnberger, K. U., Schwering, A., 2008. Analogical reasoning – A core of cognition. Künstliche Intelligenz 1, 8–12.
Gust, H., Kühnberger, K. U., Schmid, U., 2006. Metaphors and heuristic-driven theory projection (HDTP). Theoretical Computer Science 354 (1), 98–117.
Hong-Hai, D., Rahm, E., 2007. Matching large schemas: Approaches and evaluation. Information Systems 32 (6), 857 – 885.
Hummel, J. E., Holyoak, K. J., 1997. Distributed representations of structure: A theory of analogical access and mapping. Psychological Review 104 (3), 427.
Hummel, J. E., Holyoak, K. J., 2003. A symbolic-connectionist theory of relational inference and generalization. Psychological review 110 (2), 220.
Krawczyk, D. C., Holyoak, K. J., Hummel, J. E., 2004. Structural constraints and object similarity in analogical mapping and inference. Thinking & Reasoning 10 (1), 85–104.
Kuhn, W., 2001. Ontologies in support of activities in geographical space. International Journal of Geographical Information Science 15 (7), 613–631.
Lambrix, P., Tan, H., 2006. SAMBO-A system for aligning and merging biomedical ontologies. Web Semant. 4 (3), 196–206.
Loewenstein, J., Gentner, D., 2005. Relational language and the development of relational mapping. Cognitive Psychology 50 (4), 315 – 353.
Lovett, A., Tomai, E., Forbus, K. D., M. Usher, J., 2009. Solving geometric analogy problems through two-stage analogical mapping. Cognitive Science 33 (7), 1192–1231.
Lukyanenko, R., Evermann, J., 2011. A survey of cognitive theories to support data integration. In: Proceedings of the America's Conference on Information Systems (AMCIS), Detroit, MI.
Morrison, R., Krawczyk, D., Holyoak, K., Hummel, J., Chow, T., Miller, B., Knowlton, B., 2004. A neurocomputational model of analogical reasoning and its breakdown in frontotemporal lobar degeneration. Journal of Cognitive Neuroscience 16 (2), 260–271.
Morrison, R. G., Doumas, L. A. A., Richland, L. E., 2011. A computational account of children's analogical reasoning: Balancing inhibitory control in working memory and relational representation. Developmental Science 14 (3), 516–529.
Pisanelli, D., Gangemi, A., Steve, G., 2000. The role of ontologies for an effective and unambiguous dissemination of clinical guidelines. In: Knowledge Engineering and Knowledge Management Methods, Models, and Tools. Vol. 1937 of Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp. 129–139.
Raad, E., Chbeir, R., Dipanda, A., 2010. User profile matching in social networks. In: 15th International Conference on Network-Based Information Systems. IEEE Computer Society, Los Alamitos, CA, USA, pp. 297–304.
Raubal, M., Adams, B., 2010. The semantic web needs more cognition. Semantic Web 1 (1), 69–74.
Richland, L., Morrison, R., Holyoak, K., 2006. Children's development of analogical reasoning: Insights from scene analogy problems. Journal of Experimental Child Psychology 94 (3), 249–273.
Shvaiko, P., Euzenat, J., 2013. Ontology matching – state of the art and future challenges. IEEE Transactions on Knowledge and Data Engineering 25 (1), 158 –176.
Sure, Y., Staab, S., Studer, R., 2002. Methodology for development and employment of ontology based knowledge management applications. SIGMOD Record 31 (4), 18–23.
Viskontas, I., Morrison, R., Holyoak, K., Hummel, J., Knowlton, B., 2004. Relational integration, inhibition, and analogical reasoning in older adults. Psychology and Aging 19 (4), 581–591.
Wei, H., Yuzhong, Q., Gong, C., 2008. Matching large ontologies: A divide-and-conquer approach. Data & Knowledge Engineering 67 (1), 140 – 160.

Appendix A. Full Results Tables

	Encoding	Ordering	Upd	NumP	F
1	NonRel	Grouped	1	1	0.49
2	Rel	Grouped	1	1	0.52
3	NonRel	Ordered	1	1	0.44
4	Rel	Ordered	1	1	0.41
5	NonRel	OrderedCSC	1	1	0.56
6	Rel	OrderedCSC	1	1	0.60
7	NonRel	Random	1	1	0.46
8	Rel	Random	1	1	0.42
9	NonRel	Grouped	2	1	0.52
10	Rel	Grouped	2	1	0.55
11	NonRel	Ordered	2	1	0.46
12	Rel	Ordered	2	1	0.46
13	NonRel	OrderedCSC	2	1	0.50
14	Rel	OrderedCSC	2	1	0.53
15	NonRel	Random	2	1	0.56
16	Rel	Random	2	1	0.52
17	NonRel	Grouped	3	1	0.51
18	Rel	Grouped	3	1	0.54
19	NonRel	Ordered	3	1	0.41
20	Rel	Ordered	3	1	0.49
21	NonRel	OrderedCSC	3	1	0.49
22	Rel	OrderedCSC	3	1	0.56
23	NonRel	Random	3	1	0.51
24	Rel	Random	3	1	0.52
25	NonRel	Grouped	1	2	0.55
26	Rel	Grouped	1	2	0.55
27	NonRel	Ordered	1	2	0.49
28	Rel	Ordered	1	2	0.45
29	NonRel	OrderedCSC	1	2	0.60
30	Rel	OrderedCSC	1	2	0.63
31	NonRel	Random	1	2	0.50
32	Rel	Random	1	2	0.45
33	NonRel	Grouped	2	2	0.54
34	Rel	Grouped	2	2	0.57
35	NonRel	Ordered	2	2	0.50
36	Rel	Ordered	2	2	0.49
37	NonRel	OrderedCSC	2	2	0.52
38	Rel	OrderedCSC	2	2	0.55
39	NonRel	Random	2	2	0.59
40	Rel	Random	2	2	0.55
41	NonRel	Grouped	3	2	0.49
42	Rel	Grouped	3	2	0.56
43	NonRel	Ordered	3	2	0.45
44	Rel	Ordered	3	2	0.53
45	NonRel	OrderedCSC	3	2	0.48
46	Rel	OrderedCSC	3	2	0.58
47	NonRel	Random	3	2	0.53
48	Rel	Random	3	2	0.52

Table A.9: Mean of F-values for 94 OAEI tests for all experimental conditions. Encoding refers to relational or non-relational encoding, Updates is the number of propositions activated before mapping connections are updated, NumProps refers to the number each proposition is activated.

	Encoding	Ordering	Upd	NumP	F
1	NonRel	Grouped	1	1	0.56
2	Rel	Grouped	1	1	0.58
3	NonRel	Ordered	1	1	0.44
4	Rel	Ordered	1	1	0.31
5	NonRel	OrderedCSC	1	1	0.60
6	Rel	OrderedCSC	1	1	0.68
7	NonRel	Random	1	1	0.50
8	Rel	Random	1	1	0.42
9	NonRel	Grouped	2	1	0.54
10	Rel	Grouped	2	1	0.61
11	NonRel	Ordered	2	1	0.46
12	Rel	Ordered	2	1	0.47
13	NonRel	OrderedCSC	2	1	0.50
14	Rel	OrderedCSC	2	1	0.53
15	NonRel	Random	2	1	0.66
16	Rel	Random	2	1	0.56
17	NonRel	Grouped	3	1	0.54
18	Rel	Grouped	3	1	0.57
19	NonRel	Ordered	3	1	0.40
20	Rel	Ordered	3	1	0.53
21	NonRel	OrderedCSC	3	1	0.51
22	Rel	OrderedCSC	3	1	0.63
23	NonRel	Random	3	1	0.50
24	Rel	Random	3	1	0.57
25	NonRel	Grouped	1	2	0.59
26	Rel	Grouped	1	2	0.60
27	NonRel	Ordered	1	2	0.55
28	Rel	Ordered	1	2	0.38
29	NonRel	OrderedCSC	1	2	0.65
30	Rel	OrderedCSC	1	2	0.69
31	NonRel	Random	1	2	0.49
32	Rel	Random	1	2	0.43
33	NonRel	Grouped	2	2	0.62
34	Rel	Grouped	2	2	0.65
35	NonRel	Ordered	2	2	0.53
36	Rel	Ordered	2	2	0.51
37	NonRel	OrderedCSC	2	2	0.52
38	Rel	OrderedCSC	2	2	0.56
39	NonRel	Random	2	2	0.64
40	Rel	Random	2	2	0.61
41	NonRel	Grouped	3	2	0.55
42	Rel	Grouped	3	2	0.61
43	NonRel	Ordered	3	2	0.46
44	Rel	Ordered	3	2	0.53
45	NonRel	OrderedCSC	3	2	0.44
46	Rel	OrderedCSC	3	2	0.65
47	NonRel	Random	3	2	0.56
48	Rel	Random	3	2	0.55

Table A.10: Median of F-values for 94 OAEI tests for all experimental conditions. Encoding refers to relational or non-relational encoding, Updates is the number of propositions activated before mapping connections are updated, NumProps refers to the number each proposition is activated.

Appendix B. Example OAEI Alignemnt Problem

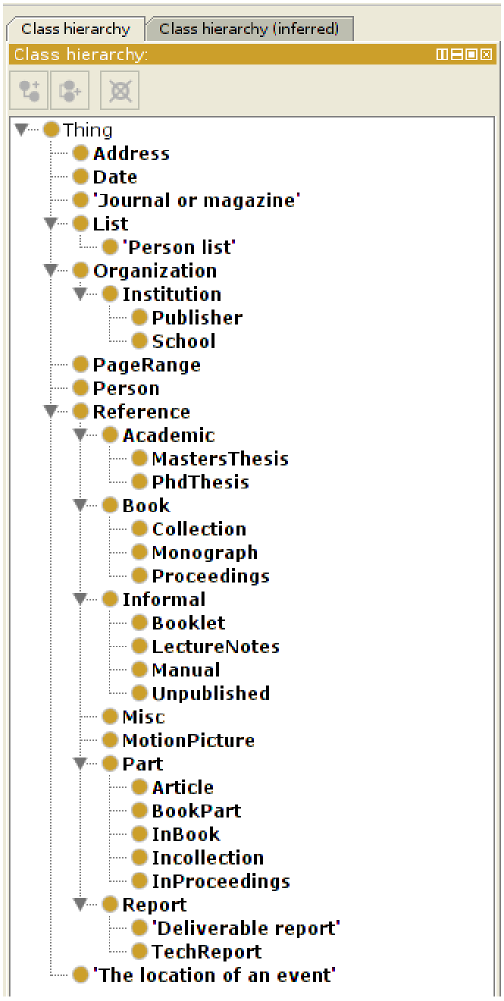

Figure B.5: Protege screenshot of the class hierarchy of the base ontology in the OAEI benchmark test suite. Test 101 aligns this ontology with itself.

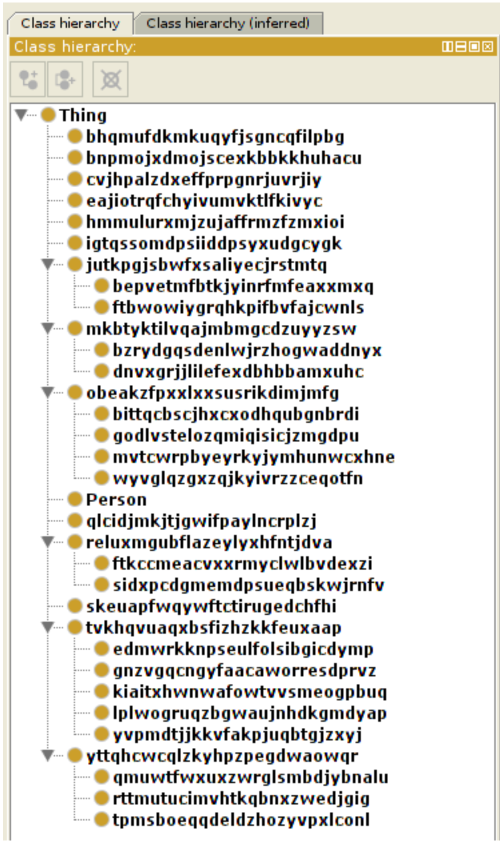

Figure B.6: Protege screenshot of the class hieararchy of the ontology for OAEI benchmark test 258. Notice the scrambling of the labels and the flattening of the class hierarchy.