# On Capturing Semantics in Ontology Mapping

**Bo Hu** and **Srinandan Dasmahapatra** and **Paul Lewis** and **Nigel Shadbolt**
University of Southampton,
Southampton SO17 1BJ, United Kingdom
{bh, sd, phl, nrs}@ecs.soton.ac.uk

**Abstract**

Ontology mapping is a complex and necessary task for many Semantic Web (SW) applications. The perspective users are faced with a number of challenges including the difficulties of capturing semantics. In this paper we present a three-dimensional ontology mapping model. This model reflects the engineering steps needed to materialise a versatile mapping system in order to meet the demands on semantic interoperability in the SW environment. We solidify the formalisation with specialised algorithms and we analyse their effectiveness and performance by way of benchmark tests.

## Introduction

Currently, research and development in the area of the Semantic Web have reached a stage that a large number of applications and services are powered by ontologies. A good wish behind the enthusiasm towards ontologies is that with shared terminologies/vocabularies comes along a community-wise consensus on the underlying semantics. Such an approach, however, has been too optimistic on two issues: i) designing a "perfect" ontology that accommodates all needs is not an easy task and ii) maintaining a consistent interpretation of an overcommitted and comprehensive ontology, even if there is one, is impractical. As a result, finding semantic equivalence among different ontologies becomes a useful alternative in coping with the semantic heterogeneity that one may be exposed to when his/her own ontology-based applications need to communicate with those from others. Ontology mapping is by no means a new research topic. It has been the focus of multidisciplinary efforts across a diverse landscape including Database, Information System, Information Retrieval (IR), etc. It has drawn more attention recently due to the aforementioned reasons and triggered enormous interest in both theoretical research and system development (Rahm & Bernstein 2001; Stumme & Maedche 2001) that gave birth to a series of conferences and workshops.

While many sophisticate tools have been developed (Benjamins *et al.* 2006), we witness a subtle shift of focus from "*how well one maps?*" towards "*how well the semantics is captured?*". Evidences of such a trend can be found in (Cruz *et al.* 2006), as many papers devoted to defining new formalisation and methods (e.g. via argumentation) to discover the "latent" semantics. In order to answer the "*how-well*" question, we cannot ignore the thousands of years of studies in philosophy w.r.t. "the meanings of words". We, hence, jigsaw similarity measures in a well-established semantic model and propose different mapping algorithms at three dimensions, i.e. the representational, conceptualisational and instance dimension. Our goal is to produce a framework that offers a full-scale treatment of semantics.



## Capturing semantics in three dimensions

A narrative interpretation of the ontology mapping problem is: "given two ontologies of the same domain (or largely overlapping domains), a set of corresponding pairs from respective ontologies should be returned to indicate how these two ontologies can be aligned". We adopt the well-known triangle model (Ogden & Richards 1923) for assigning and operationalising the meaning of ontologies (see Fig. 1). In the light of this model, an ontological entity $\mathcal{E} \in \mathcal{O}$ (where $\mathcal{O}$ is an ontology) is a 3-tuple, $\langle s, f, i \rangle$, where $s$ denotes a finite set of symbols as names, $f$ a finite set of well formed formulae as restrictions on $s$, and $i$ the *extension* of $s$ satisfying $f$. More specifically, given a set of instances $\Delta$ as the domain for interpreting $s$ and an *interpretation function* $\iota$, $i = s^{\iota}$ is a subset of $\Delta$. Meanwhile, ontology $\mathcal{O}$ is a 3-tuple $\langle \mathcal{S}, \mathcal{F}, \mathcal{I} \rangle$, where $\mathcal{S} = \{s_{\mathcal{E}} \mid \mathcal{E} \in \mathcal{O}\}$, $\mathcal{F} = \{f_{\mathcal{E}} \mid \mathcal{E} \in \mathcal{O}\}$, and $\mathcal{I} = \{i_{\mathcal{E}} \mid \mathcal{E} \in \mathcal{O}\}$. Identifying semantic equivalence is, therefore, formalised as:

**Definition 1** *Semantic equivalence between $\mathcal{E} \in \mathcal{O}$ and $\mathcal{E}' \in \mathcal{O}'$ is a mapping $\mathfrak{m} : \mathcal{E} \rightleftharpoons \mathcal{E}'$, such that, 1) $s_{\mathcal{E}} = \mathfrak{m}(s'_{\mathcal{E}'})$, and vice versa; 2) $f_{\mathcal{E}} \models \mathfrak{m}(f'_{\mathcal{E}'})$, and vice versa; and 3) $i_{\mathcal{E}} \subseteq i'_{\mathcal{E}'}$, and vice versa.*

Based on Definition 1, mapping between ontological entities $\mathcal{E}$ and $\mathcal{E}'$ is formalised as:

$$\mathtt{sim}(\mathcal{E}, \mathcal{E}') = \gamma\left(\mathtt{sim}_{\mathcal{S}}(\mathcal{E}, \mathcal{E}'), \mathtt{sim}_{\mathcal{F}}(\mathcal{E}, \mathcal{E}'), \mathtt{sim}_{\mathcal{I}}(\mathcal{E}, \mathcal{E}')\right) \quad (1)$$

where $\mathtt{sim}()$ computes the similarity between two ontological entities, $\mathcal{E}$ and $\mathcal{E}$'. That is to say that similarity between $\mathcal{E}$ and $\mathcal{E}'$ is an aggregation ($\gamma$) (e.g. weighted average) of $\mathtt{sim}_{\mathcal{S}}(\mathcal{E}, \mathcal{E}')$, $\mathtt{sim}_{\mathcal{F}}(\mathcal{E}, \mathcal{E}')$ and $\mathtt{sim}_{\mathcal{I}}(\mathcal{E}, \mathcal{E}')$ focusing on

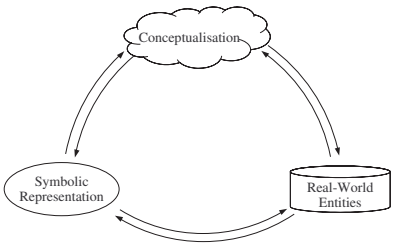


Figure 1: Meaning triangle by Ogden and Richards

the representational, conceptualisational, and instance similarity respectively.

The similarity functions, $\mathtt{sim}_{\mathcal{S}}$, $\mathtt{sim}_{\mathcal{F}}$, and $\mathtt{sim}_{\mathcal{I}}$, can be instantiated by different approaches (Kalfoglou *et al.* 2005). However, this does not suggest that capturing semantics in ontology mapping is a solved problem. Many systems claiming to offer semantic intensive mapping fail to respond to the multidimensional nature of semantics. External lexica (e.g. WordNet) and/or ontological structures have often been misread as the full "spectrum" of the semantics. Certainly, we do not disparage the importance of such information. Rather, we would like to emphasise that the external lexica and ontological structures alone only provide a limited access to the full picture of semantics which should be captured as a "trio".

## Ontology mapping as a "trio"

The three-dimensional mapping model is accommodated by algorithms that are specialised in computing the similarity w.r.t. symbolic representations, abstract conceptualisations and concrete instance data.

Similarity w.r.t. the symbolic representations is normally computed as String Distance or one of its variants (Cohen, Ravikumar, & Fienberg 2003). The algorithms aiming at the representational similarity assume that ontology engineers always know how to choose the most appropriate names for ontological entities. There are, however, no widely accepted "golden" naming standards for ontologies. Hence, methods in this family suffer from the lack of systematic measures to ensure their performance.

### Similarity of conceptualisation

A general practice of formalising conceptualisation is using logics. In this paper, we focus on the description logic (DL) based modelling languages (e.g. OWL-DL) and "dissolve" conceptualisations into semantics-bearing units which are referred to as *signatures* (Hu *et al.* 2006). We do not consider our focusing on DL-based languages as a major limitation of our approach, due to the fact that OWL, as a W3C standard, is expected to be the main language for ontologies targeting at the Semantic Web. Meanwhile, the proposed algorithm works for other modelling languages that are less expressive than OWL-DL.

**Obtaining semantic signatures** In DLs, the semantics of concepts is embedded in the logic-based constructs which need to be made explicit before computing the similarities. Explicating the embedded semantics amounts to recursively unfolding concepts till a fixed point is reached. If cyclic definitions are not allowed, i.e. no concept (property) names appear on both sides of a concept (property) introduction axiom and all definitions are in their Negation-Normal Form, i.e. the negations are applied only to concept names, it is possible to fully unfold the righthand side of all concept introduction axioms and guarantee the termination of such an unfolding process.

$$\begin{array}{ll} \mathsf{Woman} \doteq & \mathsf{Person} \sqcap \mathsf{Female} \\ \mathsf{Mother} \doteq & \mathsf{Woman} \sqcap \exists \mathsf{hasChild.Person} \\ \mathsf{MWMD} \doteq & \mathsf{Mother} \sqcap \geq_2 \mathsf{hasChild.Woman} \\ \mathsf{Female} \sqsubseteq & \top \quad \mathsf{Person} \sqsubseteq \top \end{array}$$

Figure 2: MWMD example

We adopted the construct transformation rules used in many DL systems to facilitate the concept unfolding and the signature extraction process. In Fig. 3, we present a simple example showing how MWMD (Mother with many daughters, defined in Fig. 2) is unfolded by repetitively applying the transformation rules defined for each and every DL construct (see (Baader *et al.* 2003) for further details). The unfolding process for MWMD terminates when no transformation rules are applicable.

$$\begin{aligned}
{}^{0}\Upsilon_1^{\mathsf{MWMD}} &= \{\ x : \ \mathsf{Mother} \sqcap \geq_2 \mathsf{hasChild.Woman}\ \} \\
{}^{1}\Upsilon_1^{\mathsf{MWMD}} &= \left\{ \begin{array}{ll} x : & (\mathsf{Woman} \sqcap \exists \mathsf{hasChild.Person}) \sqcap \\ & \geq_2 \mathsf{hasChild.}(\mathsf{Person} \sqcap \mathsf{Female}) \end{array} \right\} \\
{}^{2}\Upsilon_1^{\mathsf{MWMD}} &= \left\{ \begin{array}{l} x : \mathsf{Woman}, x : \exists \mathsf{hasChild.Person}, \\ x : \geq_2 \mathsf{hasChild.}(\mathsf{Person} \sqcap \mathsf{Female}) \end{array} \right\} \\
&\ldots \\
{}^{n}\Upsilon_1^{\mathsf{MWMD}} &= \left\{ \begin{array}{l} x : \mathsf{Woman}, \\ \langle x, y_1 \rangle : \mathsf{hasChild}, \ y_1 : \mathsf{Female}, \\ \langle x, y_2 \rangle : \mathsf{hasChild}, \ y_2 : \mathsf{Female}, \\ y_1 : \mathsf{Person}, \ y_2 : \mathsf{Person} \end{array} \right\}
\end{aligned}$$

Figure 3: Unfolding concept MWMD

As illustrated in Fig. 3, MWMD is completely unfolded and associated with its semantics-bearing signature, ${}^{n}\Upsilon_1^{\mathsf{MWMD}}$, that captures the semantics of the concept. Several points require further discussion. Firstly, there might be cases where concepts are defined as the union of other concepts, that are either explicitly defined elsewhere in the same ontology or introduced as anonymous ones. The unfolding rule for DL construct $\sqcup$ is non-deterministic and results in more than one signature. For instance, if we have "$\mathsf{Human} \doteq \mathsf{Man} \sqcup \mathsf{Woman}$" and define Man and Woman as "$\ldots \sqcap \forall \mathsf{hasGender.Male} \sqcap \ldots$" and "$\ldots \sqcap \forall \mathsf{hasGender.Female} \sqcap \ldots$" respectively. After unfolding, we have two signatures.

$$\begin{aligned}
{}^{n}\Upsilon_1^{\mathsf{Human}} &= \{\ldots, x : \forall \mathsf{hasGender.Male}, \ldots\} \text{ or} \\
{}^{n}\Upsilon_2^{\mathsf{Human}} &= \{\ldots, x : \forall \mathsf{hasGender.Female}, \ldots\}
\end{aligned}$$

Thirdly, universal property quantifications can only be further expanded when in the same signature, there are ele-

ments defined over the quantified property. For instance, in the above example, "$y$ : Male" is introduced if and only if both "$x$ : $\forall$hasGender.Male" and "$\langle x, y\rangle$ : hasGender" present. It is left unexpanded otherwise. This, however, does not suggest that universal quantifications over composite concepts should be treated as atomic signature elements. On the contrary, they are fully further expanded. For instance, $\forall \mathsf{R}.(\mathsf{A} \sqcap \mathsf{B})$ is transformed into $\forall \mathsf{R}.\mathsf{A} \sqcap \forall \mathsf{R}.\mathsf{B}$.

The unfolding process stops when a fixed point is reached, i.e. ${}^{n}\Upsilon = {}^{(n-1)}\Upsilon$. As demonstrated in (Horrocks & Sattler 2005), by carefully selecting a set of admissible constructs termination is guaranteed w.r.t. acyclic ontologies.

**Computing similarity** After signaturing all concepts in $\mathcal{O}$, a set of the most basic building blocks of $\mathcal{O}$ is acquired as the elements of concept signatures. It is evident that a formula that can be derived from $\mathsf{C} \in \mathcal{O}$ w.r.t. $\mathcal{O}$ is a subset of $\Upsilon_i^{\mathsf{C}}$. The conceptual similarity, therefore, can be obtained by counting the number of the subsets of the intersection of the respective signatures. When concepts are unfolded into more than one signature due to the presence of disjunctions, the maximum similarity is computed among the alternatives. Let $\Upsilon_i$ and $\Upsilon'_j$ be the signatures of $\mathcal{E}$ and $\mathcal{E}'$ respectively and $|\cdot|$ give the set cardinality, we have

$$\mathtt{sim}_{\mathcal{F}}(\mathcal{E}, \mathcal{E}') = \max_{(\Upsilon_i \text{ of } \mathcal{E}, \Upsilon'_j \text{ of } \mathcal{E}')} \frac{2 \cdot 2^{|\Upsilon_i \cap \Upsilon'_j|}}{2^{|\Upsilon_i|} + 2^{|\Upsilon'_j|}}$$

If $\mathcal{E}$ and $\mathcal{E}'$ are from different ontologies, their signature elements might not be comparable to startup the similarity computation. It is necessary to align the respective signature elements first in order to compute $\Upsilon \cap \Upsilon'$: if $\phi$ denotes the set of one-to-one alignments between $\Upsilon$ and $\Upsilon'$, $|\Upsilon \cap \Upsilon'| = |\ \phi\ |$. In practice, primitive concepts and properties normally do not provide many clues for obtaining $\phi$ apart from their names and, in some cases, their instances. When computing the conceptualisational similarity, we want to avoid relying on syntax-based name similarities, even for bootstrapping the whole process. We, therefore, either introduce the alignments manually or allow them to gradually emerge as follows.

Let $\Upsilon_i^{\mathsf{C}}$ be the $i$th signature of an arbitrary concept $\mathsf{C} \in \mathcal{O}$, we define

$$\mathfrak{F} = \bigcup_{\mathsf{C} \in \mathcal{O}} \left( \bigcup_{\Upsilon_i^{\mathsf{C}} \text{ of } \mathsf{C}} \Upsilon_i^{\mathsf{C}} \right)$$

as the set of all elements that appear in any signatures of $\mathcal{O}$. If $\Phi : \mathfrak{F} \times \mathfrak{F}'$ gives all possible alignments between $\mathfrak{F}$ of $\mathcal{O}$ and $\mathfrak{F}'$ of $\mathcal{O}'$, we repetitively examine $\phi \in \Phi$ to maximise

$$\arg\max_{\phi \in \Phi} \sum_{\mathcal{E} \in \mathcal{O}, \mathcal{E}' \in \mathcal{O}'} \mathtt{sim}_{\mathcal{F}}^{\phi}(\mathcal{E}, \mathcal{E}') \qquad (2)$$

The $\phi$ maximising Equation 2 is the recommended alignment between signature elements, based on which similarities of concepts are computed. For very large ontologies, checking each pair of concepts to maximise Equation 2 might not be feasible. In practice, we reduce the search space of Equation 2 with two relaxations: i) we examine only leaf concepts, i.e. concepts without children, and ignore interim ones to reduce the computational burden of each iteration and ii) we align only compatible signature elements. More specifically, we distinguish three types of signature elements: concept-type (e.g. "$x$:Woman"), property-type (e.g. "$\langle x, y_1\rangle$:hasChild"), and elements featured by universal property quantifications (e.g. "$x$:$\forall$hasChild.Person"). Moreover, for each type, we only compare elements at the same nesting property reference level. For instance, in the previous example, we do not compare $x$ : Woman $\in {}^{n}\Upsilon_1^{\mathsf{MWMD}}$ against $y_1$ : Female $\in {}^{n}\Upsilon_1^{\mathsf{HM}}$ due to the fact that the latter is introduced in $\Upsilon_1^{\mathsf{HM}}$ through property hasChild and thus is at a different nesting level from the former. In many ontologies, compared to fully defined concepts, the number of primitive concepts and properties is relatively small. So is size of $\mathfrak{F}$. For instance, FMA Ontology[1] has roughly 70k concepts with only approximately 100 properties and 20 primitive concepts.

## Similarity of instances

Many approaches, either acting as a stand alone tool (Stumme & Maedche 2001) or being a building block of a large framework (Doan *et al.* 2002), have been proposed to discover the meanings of concepts from the instance data and subsequently semantic equivalences therein. When a relatively complete set of instances can be identified w.r.t. a populated ontology, the semantics of ontological entities can be reflected through the distribution of such instances. A major assumption made by techniques belonging to this family is that concepts with similar semantics share instances (Doan *et al.* 2002) and an understanding of their distribution contributes to the understanding of the semantics. Data Mining and IR techniques are frequently employed to winnow away apparent discrepancies as well as discover the hidden correlations among instances. Some recent efforts along this line of research are discussed in (Kalfoglou *et al.* 2005).

However, the ideal situation assumed by the authors of (Doan *et al.* 2002) is not always true. Many ontologies published on the loosely regulated internet might only have a skewed set of instances or not have any instances at all. We, therefore, developed algorithms to tackle ontologies without sufficient instance level information.

**When instance data is partially available** Although many automated learning methods have been proposed (Cruz *et al.* 2006), manually populating ontology is still dominant. The lack of human labour and proper domain knowledge, therefore, results in many under-populated ontologies in the Semantic Web. We differentiate two types of under-populated ontologies: those with at least one instance asserted for each leaf concept and those with un-instantiated leaf concepts. Ontologies falling in the latter category are treated in the same way as unpopulated ones while those belonging to the first category is discussed in this section.

The fundamental idea is that the entire internet presents it-

[1] http://sig.biostr.washington.edu/projects/fm/index.html

Generate web search query, `Query`

**Require:** 1) concept hierarchy $\mathcal{H}(\prec)$ of ontology $\mathcal{O}$
2) concept $\mathsf{C}$, $\mathsf{D}$, $\mathsf{E}$ and instance $I_i$ in $\mathcal{O}$
3) $\alpha \in \{\geq_n, \leq_n, =_n, \forall, \exists\}$

**function** `Query`($\mathsf{C}$)
$\mathcal{Q} = \{$*names in* $\mathsf{C}\}$
*% extend $\mathcal{Q}$ with names from parent concepts %*
**if** "*conjunction of* $\mathsf{D}$ *and* $\mathsf{E}$" $\prec \mathsf{C} \in \mathcal{H}$ **then**
$\mathcal{Q} = \{$`Query`($\mathsf{D}$), "AND" + `Query`($\mathsf{E}$)$\} \cup \mathcal{Q}$
**else if** "*disjunction of* $\mathsf{D}$ *and* $\mathsf{E}$" $\prec \mathsf{C} \in \mathcal{H}$ **then**
$\mathcal{Q} = \{$`Query`($\mathsf{D}$), "OR" + `Query`($\mathsf{E}$)$\} \cup \mathcal{Q}$
**else if** "*complement of* $\mathsf{D}$" $\prec \mathsf{C} \in \mathcal{H}$ **then**
$\mathcal{Q} = \{$"NOT" + `Query`($\mathsf{D}$)$\} \cup \mathcal{Q}$
**else if** "$\alpha$ $\mathsf{R}$ $\mathsf{D}$" $\prec \mathsf{C} \in \mathcal{H}$ **then**
$\mathcal{Q} = \{$"AND" + `Query`($\mathsf{D}$)$\} \cup \{$"AND" + $\mathsf{R}\} \cup \mathcal{Q}$
**else if** $\mathsf{D} \prec \mathsf{C} \in \mathcal{H}$ **then**
$\mathcal{Q} = \{$`Query`($\mathsf{D}$)$\} \cup \mathcal{Q}$
**end if**
**return** `cleanup`($\mathcal{Q}$)
**end function**

Figure 4: Query generation of $\mathsf{C}$

self as a very large corpus from which we can extract, assess, and assert instances for under-populated ontologies. This is realised by generating internet search queries and pulling out web pages as candidate *virtual* instances which are then subject to further verification. The vast internet presents itself as both an opportunity and a challenge. It is an opportunity in the sense that in line with the vision of emergent semantics (Staab *et al.* 2002), the Internet provides a corpus with ordinary web users constantly and collaboratively pooling in their knowledge and view of the world. The accumulative effect is not trivia and can be exploited to discover the latent semantics among terms. The challenge is equally distinct. The sheer size of the *virtual* instance population denies the possibility of developing time-efficient tools to investigate individual web pages, assess them and treat them in the same way as ordinary instances. When all the leaf concepts are instantiated, these properly defined instances—even though not complete—can be treated as clues to validating *virtual* instances. This process is summarised as follows: Let $\mathcal{E} \in \mathcal{O}$ and $\mathcal{E}' \in \mathcal{O}'$ be two concepts, $i_{\mathcal{E}}$ and $i'_{\mathcal{E}'}$ be the properly defined instances of $\mathcal{E}$ and $\mathcal{E}'$ respectively.

1. For every $\mathcal{E} \in \mathcal{O}$, we generate a search query according to the algorithm in Fig. 4 where we assume that concepts in $\mathcal{O}$ are arranged in a hierarchical structure $\mathcal{H}$ regulated by $\prec$, i.e. $\mathsf{D} \prec \mathsf{C}$ implies that $\mathsf{D}$ precedes $\mathsf{C}$ in $\mathcal{H}$. `cleanup`() performs stop-word removal, tokenisation, and other NLP techniques.
2. The generated queries are fed into an internet search engine with the capacity of handling basic logic operators, e.g. Google™ Search API[2]. We pool the top $k$ retrieved pages and denote pages retrieved with a particular query as $\omega_{\mathcal{E}} = \textit{hit}\,(\texttt{Query}(\mathcal{E}))$.
3. $\omega_{\mathcal{E}}$ is refined by: i) computing the distance between the defined instance $i_{\mathcal{E}}$ on the one hand and every candidate *virtual* instance $vi \in \omega_{\mathcal{E}}$ on the other hand and ii) pruning *vi*s with distance greater than the threshold $t_1$.
4. Repeat Step 1-3 on every $\mathcal{E}' \in \mathcal{O}'$ with the same $k$ and $t_1$.
5. For refined $\omega_{\mathcal{E}}$ and $\omega_{\mathcal{E}'}$, we compute the distance $d$ between every $vi' \in \omega_{\mathcal{E}'}$ and $i_{\mathcal{E}}$: if $d \geq t_2$, we say $vi'$ can be classified against $\mathcal{E}$. The same procedure is performed on $\omega_{\mathcal{E}}$ w.r.t. $i'_{\mathcal{E}'}$ of $\mathcal{E}'$. Non-leaf concepts acquire *virtual* instances automatically from their sub-concepts.
6. Based on the classified *virtual* instances, we compute the similarity $\texttt{sim}_{\mathcal{I}}(\mathcal{E}, \mathcal{E}') = |\omega_{\mathcal{E}} \cap \omega_{\mathcal{E}'}| \,/\, |\omega_{\mathcal{E}} \cup \omega_{\mathcal{E}'}|$.

Many methods can be used to acquire the distance between *virtual* and "real" instances. A straightforward approach is to exploit the well-established *bag of words* model converting both types of instances as vectors of words and compute the cosine distance of vectors (van Rijsbergen 1979). A few factors may impinge on the performance of the above method. Firstly, ideally the cut-off value $k$ and threshold $t_1$ and $t_2$ should be repetitively evaluated till an optimal setting is researched. In our initial implementation, however, we empirically assigned $k$=100 and $t_1$=$t_2$=$3/4\mu$ where $\mu$ is the arithmetic mean distance between the *virtual* instances and the defined one for a particular concept. Secondly, when a leaf concept has more than one defined instance, we randomly pick up one against which *virtual* instances are examined. Further work should conceive a "clever" mechanism to prune unqualified *virtual* instances based on information drawn from all "real" instances of a concept.

**When instances are not available** There are occasions that one needs to map completely unpopulated ontologies. By taking full advantage of the Internet search engines, one could query to retrieve web pages indexed with a set of words. Although such *virtual* instances cannot be verified, instead of using them directly, we can discover the semantic correlations among them by leveraging statistic methods, e.g. *normalised google distance* (Cilibrasi & Vitanyi 2007).

One way to explore the semantic correlation is to compute the conditional probability between concepts, $P(\mathsf{C} \mid \mathsf{D}) = P(\mathsf{C}, \mathsf{D})/P(\mathsf{D})$. Given two arbitrary concepts $\mathsf{C}$ and $\mathsf{D}$, their joint probability is approximated as

$$P(\mathsf{C}, \mathsf{D}) \approx \frac{|\ \textit{hit}\,(\texttt{Query}(\mathsf{C})\ \text{AND}\ \texttt{Query}(\mathsf{D}))\ |}{\Omega}$$

where $\Omega$ is the total number of web pages that the selected search engine currently indexes and $|\textit{hit()}|$ gives the number of web pages returned by the search engine. The applicability of using internet search engines to discover meanings of general terms/concepts has been demonstrated in (Cilibrasi & Vitanyi 2007). Ostensibly, the Internet, as a repository of general information, does not convey many useful messages for specific domains, e.g. military domains and medical domains. Our experience, however, confirmed otherwise. Although much fewer hits are returned comparing to general terms, the results are still statistically significant.

The similarity $\texttt{sim}_{\mathcal{I}}(\mathcal{E}, \mathcal{E}')$ is then computed as:

$$\texttt{sim}_{\mathcal{I}}(\mathcal{E}, \mathcal{E}') = \frac{1}{2} \cdot \left( \frac{P(\mathcal{E}, \mathcal{E}')}{P(\mathcal{E}')} + \frac{P(\mathcal{E}, \mathcal{E}')}{P(\mathcal{E})} \right)$$

[2]http://www.google.com/apis/index.html

## Evaluation

Mapping algorithms of the "trio" model discussed in the previous section are implemented and their performance is evaluated against benchmark tests from the Ontology Alignment Evaluation Initiative (OAEI)[3]. OAEI tests contain 50 different cases each focusing on one or a combination of several factors that might differentiate ontologies. They are particularly useful in our case, because there are dedicated cases for representational and conceptualisational variances.

### Details of test case

The OAEI reference ontology (test case 101) contains 33 concepts and 30 properties from the bibliography domain. The variances w.r.t. concepts, properties and the hierarchical structure are introduced in different test cases. Roughly, the OAEI 2006 benchmark tests can be categorised into the following groups:

**103-104** ontologies codified in different modelling languages (e.g. OWL/XML/RDF) while both the representation and conceptualisation remain the same as the reference ontology *101*;

**201-210** ontologies with only linguistic variances, e.g. concept names replaced with random strings, translated into foreign languages, generalised using hypernyms, etc.;

**221-247** ontologies with conceptualisational variances, e.g. flattened hierarchies, suppressed properties, suppressed instances, etc.;

**248-266** ontologies with both representational and conceptualisational variances, i.e. the combinations of changes from case group *201-210* and *221-247*;

**301-304** Real-life bibliographic ontologies.

Among them, the cases belonging to 300 family are of particular interest because they present both representational and conceptualisational differences that reflect the natural variances among independent ontology engineers.

### Comparing mapping results

The organisers of OAEI contest have crafted reference alignments for every test case. If taking such alignments as the ground truth, it is possible to compute the *precision*, *recall* and *f-measure* and quantitatively analyse the strength and weakness of our algorithms (see Table 1 for the results). When evaluating our similarity algorithms, we select only the best candidates returned by such algorithms and ignore all other less favourite ones. That is to say that we select the candidate that has the highest similarity or highest rank and produce exactly one mapping from the target ontology for every concept of the source ontology.

*Precision* ($p$) is, therefore, computed as the fraction of the reference compliant mappings in the list of all alignments returned by our algorithms. *Recall* ($r$) is the fraction of the reference compliant mappings returned by our algorithms in the entire set of reference alignments. *F-measure* ($f$) presents a combined criteria of *precision* and *recall* as $f = 2p \cdot r/(p + r)$.

[3]http://oaei.ontologymatching.org/2006/benchmarks/

### Discussions

As expected, representational similarity methods are sufficient for cases that only have syntactical variances, e.g. benchmark test 103 and 104. $Sim_{\text{rep}}$ also performs well when only conceptualisational variances present, e.g. test group *221-247* as illustrated in Fig. 5, but it is still unclear whether or not two concepts should be considered equivalent if their conceptualisations are different. Conceptualisation-based method $Sim_{\text{con}}$ outperforms $Sim_{\text{rep}}$ when concept identifiers/names cannot provide enough information for aligning, e.g. benchmark test group *201-210*. For instance, in test 201, both concept and property names are replaced with random strings—in which case we encountered the worst case scenario for $Sim_{\text{con}}$ as well: every possible combination of the signature elements from the source and target ontologies is evaluated resulting in a total number of 6000+ iterations for a small ontologies with only 33 concepts and roughly 40 properties. Such an extreme scenario, however, seldom occur in real-life mapping tasks, as concept and property names always give clues for possible alignments between signature elements. In benchmark test group 248-266, after examining all potential alignments, our algorithm eventually converged on one set of alignments between signature elements resulting in concept and property mappings with relatively high precision and recall: 0.87 and 0.52 respectively.

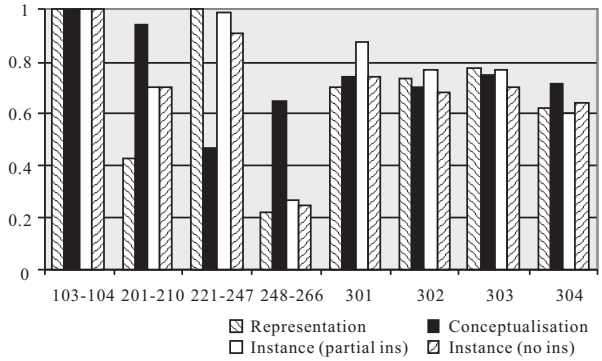


Figure 5: representation v.s. conceptualisation v.s. instance

In real-life tests, both representational and conceptualisational similarities are limited by the naming and modelling conventions and thus algorithms focusing on one aspect do not have distinguishable advantages over those based on another (test 301 to 304 in Fig. 5). As an alternative, instance data reveal the intended meaning of concepts and thus can complement the algorithms based on representational and conceptualisational characteristics. Due to the unavailability of properly populated ontologies—many benchmark tests are not properly populated, we evaluated only the two algorithms for under-populated ontologies. In order to facilitate the verification process against *virtual* instances, when running the algorithm targeting at partially populated ontologies ($Sim_{\text{par-ins}}$), we manually introduced one instance for every un-instantiated leaf concept. Benefiting from the carefully crafted exemplar instances, $Sim_{\text{par-ins}}$ achieved good results in the majority of the test cases (see Fig. 5). We would argue that the use of exemplar instances in not a disadvantage. In many real-life scenarios, ontologies are used for acquiring

| | 103-104 | 201-210 | 221-247 | 248-266 | 301 | 302 | 303 | 304 |
|---|---|---|---|---|---|---|---|---|
| | ① Syntax | ② Name | ③ Structure | ② + ③ | Real-life ontologies | | | |
| Representation ($Sim_{\text{rep}}$) | 1.00 | 0.57 | 1.0 | 0.22 | 0.70 | 0.73 | 0.78 | 0.62 |
| Conceptualisation ($Sim_{\text{con}}$) | 1.00 | 0.94 | 0.47 | 0.65 | 0.74 | 0.70 | 0.75 | 0.71 |
| Partial instance ($Sim_{\text{par-ins}}$) | 1.00 | 0.70 | 0.99 | 0.27 | 0.88 | 0.77 | 0.77 | 0.60 |
| No instance ($Sim_{\text{no-ins}}$) | 1.00 | 0.70 | 0.91 | 0.25 | 0.74 | 0.68 | 0.70 | 0.64 |

Table 1: F-Measure of each category of test cases with different types of variances

and storing knowledge. Hence, when a mapping task is requested, we expect that the ontology owners might already have a partially populated ontology and/or possess enough domain knowledge to manually instantiate all leaf concepts. This assumption, of course, need to be validated by further studies. Moreover, although both algorithms have achieved at least an average f-measure of those of the representational and conceptualisational similarity measures, the benchmark test is not large and diverse enough to conclude that the *virtual* instance based methods perform equally well in other real-life mapping cases.

## Conclusion and future work

In this paper, we adopted the triangle metaphor by Ogden and Richards (Ogden & Richards 1923) and presented a "trio" model for exploring the full-scale semantics in ontology mapping. At each vertex (dimension) of the meaning triangle, we proposed and implemented algorithms that can best represent the alike and evaluated their performance by means of ontology mapping benchmark test. Our experiment demonstrated that algorithms focusing on the different aspects of the semantics can be complementary.

Of course, revealing the true meanings is always one of the most controversial issues when coping with semantic heterogeneity. There is irreconcilable argument on what the semantics is as well as how semantics can be revealed. In practice, surprisingly, representation-only methods have manifested their efficiency in special settings through many industry-strength mapping/aligning systems (Noy, Doan, & Halevy 2005). Such a fact, however, should not be interpreted as a counterexample of the advocation of comprehensive similarity measures across different semantic dimensions. We would argue that concept and property names only provide evidences on one aspect of the semantics. They need to be combined with and complemented by other information. To this front, we have developed the most complete framework so far. Although our framework achieved good results w.r.t. the purposely designed benchmark test, its applicability and performance need to be further investigated and demonstrated through large-scale, real-life ontologies and optimised accordingly.

## Acknowledgements

This work is supported under the OpenKnowledge STREP projects funded by EU Framework 6 under Grant number IST-FP6-027253.

## References

Baader, F.; Calvanese, D.; McGuinness, D.; Nardi, D.; and Patel-Schneider, P., eds. 2003. *The Description Logic Handbook: Theory, Implementation and Applications*. Cambridge University Press.

Benjamins, R.; Euzenat, J.; Noy, N.; Shvaiko, P.; and Stuckenschmidt, H. adn Uschold, M., eds. 2006. *Proc. of the Ontology Matching Workshop*.

Cilibrasi, R., and Vitanyi, P. 2007. The Google Similarity Distance. *IEEE Trans. on Knowledge and Data Engineering* (3):370–383.

Cohen, W.; Ravikumar, P.; and Fienberg, S. 2003. A comparison of string distance metrics for name-matching tasks. In *Proc. of the IIWeb*, 73–78.

Cruz, I.; Decker, S.; Allemang, D.; Preist, C.; Schwabe, D.; Mika, P.; Uschold, M.; and Aroyo, L., eds. 2006. *Proc. of the ISWC 2006*.

Doan, A.; Madhavan, J.; Domingos, P.; and Halevy, A. 2002. Learning to map between ontologies on the semantic web. In *Proc. of WWW02*, 662–673.

Horrocks, I., and Sattler, U. 2005. A tableaux decision procedure for $\mathcal{SHOIQ}$. In *Proc. of the 21st IJCAI*.

Hu, B.; Kalfoglou, Y.; Alani, H.; Dupplaw, D.; Lewis, P.; and Shadbolt, N. 2006. Semantic metrics. In *Proc. of the 15th EKAW*, 166–181.

Kalfoglou, Y.; Hu, B.; Reynolds, D.; and Shadbolt, N. 2005. Semantic integration technologies. 6th month deliverable, University of Southampton and HP Labs.

Noy, N.; Doan, A.; and Halevy, A. 2005. Special Issue on semantic integration. In *AI Magazine*, volume 26.

Ogden, C., and Richards, I. 1923. *The Meaning of Meaning: A study of the influence of language upon thought and of the science of symbolism*. Harcourt Brace Jovanovich. 1989 republished.

Rahm, E., and Bernstein, P. 2001. A survey of approaches to automatic schema matching. *The VLDB Journal* 10:334–350.

Staab, S.; Santini, S.; Nack, F.; Steels, L.; and Maedche, A. 2002. Emergent semantics. *Intel. Sys., IEEE* 17(1):78–86.

Stumme, G., and Maedche, A. 2001. FCA-Merge: Bottom-Up Merging of Ontologies. In *Proc. of the 17th IJCAI*, 225–230.

van Rijsbergen, C. J. 1979. *Information Retrieval*. Dept. of Computer Science, University of Glasgow, 2nd edition.