Handling interoperability by learning ontology mappings

by Nico Roos and Floris Wiesman

Report MICC 06-04

Universiteit Maastricht
MICC/IKAT
P.O.Box 616
6200 MD Maastricht
The Netherlands

Maastricht, 2006 ISSN 0922-8721

Handling interoperability by learning ontology mappings

Nico Roos[*] Floris Wiesman[†]

MICC Technical Report 06-04
Universiteit Maastricht
ISSN 0922-8721

Abstract

Communication in open systems is hampered by the lack of standardization of the ontology used to describe information. This problem is known as the *interoperability* problem. Establishing a mapping between pairs of ontologies is a practical alternative for formulating standard ontologies. Of course, creating such a mapping by hand is not applicable in an open environment such as the Internet. Therefore, this paper proposes a domain independent method for learning a mapping between ontologies.[1] The learning method is based on exchanging instances of concepts that are defined in the ontologies. The method starts with identifying pairs of instances of concepts that describe the same entity in the world, followed by proposing and evaluating mappings between the ontologies using the pairs of instances. For each step of this method, a measure based on relative probabilities has been derived and is used in the decision process. Important benefits of the method are that (a) no domain knowledge is required, and (b) the structures of ontologies between which a mapping must be established, play no role. Experimental results show that the proposed method gives high quality mappings between ontologies.

1 Introduction

The Internet offers new possibilities for accessing and sharing information. The semantic web aims at sharing information in a machine interpretable way, agent technology aims at enabling collaboration between distributed, autonomous and possibly self-interested programs, federated database systems aim at transparently integrating multiple autonomous database systems into a single federated database, and electronic data exchange aims to reduce economic costs of, for instance, supply chains. In each of these areas accessing and sharing of information is hampered by a lack of standardization of the way information is represented. This is called the *interoperability* problem [1, 10, 16, 17, 19, 28, 30]. The cost of the interoperability problem is considered to be significant. The National Institute of Standards and Technology (USA) estimates the costs of the interoperability problem to be about $15.8 billon annually for the US capital facilities industry [6].

The lack of standardization causing the interoperability problem concerns the way information can be accessed (*structural heterogeneity*) and the interpretation of the stored information (*semantic heterogeneity*). We assume that structural heterogeneity will not be an issue. The semantic web and electronic data exchange are based on standards formats such as OWL and XML, and communication between is based on standard languages such as ACL and KQML. Structural heterogeneity can be an issue in federated database systems.

Semantic heterogeneity concerns the interpretation of the communicated information, especially the meaning of concepts and relations used in the communication. *Ontologies* [8] are intended to provide a standard vocabulary, and to provide semantics through the specification of logical relations between concepts and relations. Therefore, the development of standard ontologies will

[*]Universiteit Maastricht, Institute for Knowledge and Agent Technology, P.O.Box 616, 6200 MD Maastricht, The Netherlands. email: roos@cs.unimaas.nl

[†]Department of Medical Informatics, Academic Medical Center, University of Amsterdam, Meibergdreef 9, 1105 AZ Amsterdam, The Netherlands. email: f.j.wiesman@amc.uva.nl

[1]The presented method is an improvement of the method described in [31, 32].

offer a solution for the interoperability problem. Examples of standard ontologies are the Dublin Core [29] and the ontologies of the Ontolingua library [4, 7]. Moreover, within the semantic web community, languages such as DAML+OIL and OWL [15] have been developed for specifying ontologies.

The development of standard ontologies is expected to take a very long time because of the number of topics on which a standard ontology must be developed is almost infinite. Moreover, different conceptualizations of a domain are possible depending on the application domain and the developers' point of views.

Instead of developing standard ontologies, one can also use mappings between pairs of ontologies to enable information exchange. Creating such a mapping by hand is not feasible in an open environment. Therefore, this paper proposes a domain independent method for learning a mapping between ontologies. In short, the method consists of two steps: (*i*) identifying pairs of instances of concepts that describe the same entity in the world and (*ii*) proposing and evaluating mappings between the ontologies using the pairs of instances. Before presenting our method for learning an ontology mapping in more detail, in Section 2 we will first discuss the interoperability problem in more detail, and in Section 3 we point out some problems in current approaches. Section 4 outlines the basic ideas of our learning method while Section 5 presents experiments with learning ontology mappings. Section 6 discusses some extensions of the presented approach and Section 7 concludes the paper.

2 Heterogeneity

In order to reach interoperability, two problems must be dealt with, namely: *structural heterogeneity* and *semantic heterogeneity* [21, 28]. Structural heterogeneity concerns the different representations of information. Information described by the same ontology can be represented in different ways. This is a problem for heterogeneous databases but not for agents or the semantic web.

Semantic heterogeneity concerns the intended meaning of the described information. Information about, for instance, persons can be described by different ontologies. We distinguish the following differences between ontologies[2]:

1. *Structural conflicts* arise where different semantic structures are used [1].

2. *Naming conflicts* arise where different names for the same type of information are used, or the same name for (slightly) different types of information [1, 28].

3. *Data conflicts* arise where different representations exist of the same data [12]. These conflicts can be refined in conflicts because of *different units*, conflicts because of *different precision*, and conflicts because of *different expressions* (e.g., using 'van der Belt' or 'Belt, van der' to describe a person's family name).

The two ontologies shown in Figures 1 and 2 illustrate some forms of semantic heterogeneity. Both ontologies define a concept 'person' in terms of relations/subconcepts and attributes. In these simple ontologies, the attributes are located on the right hand side and the concepts are printed in italics. Note that Ontology 1 contains a 'father' and a 'mother' relation with an instance of the concept 'person'. The dashed line denotes the definition of the concept 'person' with which a person has a 'father' and a 'mother' relation.

In Ontology 1, 'street' also includes the house number, and 'phone number' describes the country code, the area code, and the local number. In Ontology 2, 'phone number' only describes the local number. The 'area code' and the 'country code' are stored with the city and the country, respectively.

Each ontology clearly has a different structure. Ontology 1 is flat while Ontology 2 has a hierarchical structure. This structural conflict can be solved relatively easy because Ontology 2 more or less extends Ontology 1. When the two ontologies have completely different hierarchical structures, the structural conflict becomes a more serious.

The naming conflicts between the two ontologies form a more severe problem. Different concept names are used for the same type of data; e.g., 'first name' and 'christian name'. Moreover, the

[2]A slightly different categorization of differences between ontologies is given in [27]

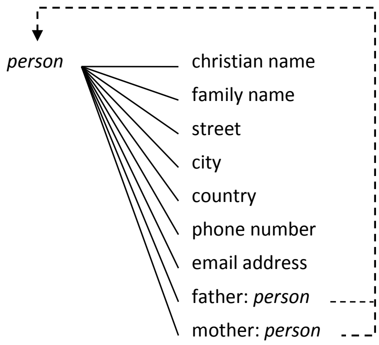

Figure 1: Ontology 1

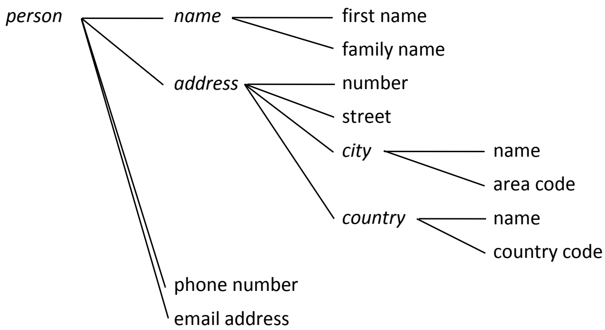

Figure 2: Ontology 2

same concept name is used for slightly different types of data; e.g., ‘street’. In Ontology 1 ‘street’ denotes both the *street name* and the *house number* while in Ontology 2 it only denotes the *street name*. Hence, in order to reach interoperability, we must be able to *split* and *concatenate* data fields. For example, an instance of the concept ‘street’ in Ontology 1 containing the value ‘Castle Lane 1’ must be split into ‘Castle Lane’ and ‘1’ in order to map ‘street’ in Ontology 1 to ‘street’ and ‘number’ in Ontology 2. The inverse mapping requires concatenating ‘Castle Lane’ and ‘1’.

In Ontology 2, the concept ‘person’ does not contain ‘father’ or ‘mother’ relations with an instance of ‘person’. If such a relation would be present, we would not only like to learn a mapping between concepts but also a mapping between relations, such as the ‘father’ and ‘mother’ relation, relating two concepts.

We can conclude that to reach interoperability we have to find a mapping from the concepts of one ontology to the concepts of another ontology using concatenating and splitting operators in the mapping process. Moreover, we have to find a mapping from the relations between the concepts of one ontology to the relations between concepts of the other ontology.

3 Related work

To deal with semantic heterogeneity, several solutions have been proposed. Many of the proposed solutions try to derive a common ontology by some (semi) automatic process, see for instance [1, 10, 16, 19, 30]. These approaches heavily rely on assumptions such as:

- Concepts are defined using a set of shared attributes,
- Different ontologies are the result of differentiations of one initial ontology,
- A human specifies relations between concepts of different ontologies and resolves possible conflicts.

Unfortunately these assumptions often cannot be met. Instead of deriving a common ontology, other approaches try to establish a mapping between two ontologies directly [2, 3, 5, 13, 17, 18, 22, 25, 26].

Firat et al. [5] assume that concepts of a shared ontology have slightly different interpretations in different contexts. They propose the use of a meta-ontology to describe the context dependent interpretations using 'type-modifiers'; e.g. a type-modifier to denote that profit is interpreted as 'profit before taxes' or 'profit after taxes'. These type-modifiers enable them to derive a mapping and also handle data conflicts caused by different units and different precision.

Papazoglou et al. [17] assume that the same naming conventions are used in different databases and that for each database an abstract description model describes the types of relations that hold between concepts that are specified. The possible relation types are common knowledge. From this information a mapping between the databases can be derived. The disadvantage of this approach is that it cannot handle naming conflicts.

Van Eijk et al. [26] use first order logic to give a characterization of establishing a mapping. They assume a multi-agent system in which agents communicate using first order formulas. A mapping consists of a set of translation formulas each expressing an equivalence between expressions. The disadvantage of this approach is that it cannot handle structural conflicts, and naming conflicts. Moreover, they do not provide a method for establishing a mapping.

Burnstein et al. [2] have proposed a somewhat similar approach based on λ-calculus. Their approach does provide a method for establishing a mapping. They assume, however, the existence of a common 'topic domain' in which the meaning of terms is specified.

Haase and Motik [9] also propose a similar approach for integrating OWL ontologies. They assume that an ontology integrating a set of source ontologies is given together with mapping rules for each source ontology.

In [31, 32], the authors have outlined the initial ideas behind their learning method. Likelihood measures for mappings were derived, which, in combination with threshold values, were used to select a mapping [31]. Additional experiments with the described approach demonstrated that the optimal thresholds values depended on the ontologies between which a mapping had to be established. This dependency was caused by an invalid assumption underlying the derived measures. This report resolves this problem by presenting new measures. Moreover, this reports extends the results described in [31].

Prasad et al. [20] have adapted the method for learning a mapping between ontologies describe in [32] to document classification. Instead of instances of concepts known in both ontologies, Prasad et al. classify documents in both ontologies and subsequently update the probability that a candidate mapping is correct. A drawback is that the candidate mappings have to be given in advance.

Several other approaches apply machine learning to learn classifiers for concepts [3, 13, 22]. Doan et al. [3] (GLUE) and Lacher & Groh [13] use the classifiers to estimate the (joint) probability that a two concepts are similar. Doan et al. define a similarity measure based on the Jaccard coefficient using the joint probability, while Lacher & Groh define a similarity measure based on the difference between probabilities of two pairs. Both similarity measures, however, do not give us the probability that a mapping is correct.

The classifiers that are learned in the approach of Soh [22] are concept classification rules. Concept translations are derived by matching the classification rules.

Lee et al. [14] propose a three step approach; (1) humans specifying semantic types for concepts, (2) filtering possible mappings between concepts based on the semantic types and (3) determining mappings between attributes and concept using a ranking based on the correspondence of names, data types and the number of corresponding relations. For the latter ranking only an intuitive under pinning is given.

4 Learning ontology mappings

We focus on establishing a mapping between two concepts, one in each ontology. No restrictions are placed on the structure of a concept. A concept may be defined as an aggregation of attributes and sub-concepts. This aggregation may even, directly or indirectly, contain the concept that is being defined. That is, we do not, for instance, exclude a concept 'person', the description of which contains a 'father' and 'mother' relation with the concept 'person'.

We will discuss learning an ontology mapping in terms of agents that wish to exchange information. This choice does not limit the proposed approach in any way to the area of agent technology. Instead of communicating agents, one may also consider two programs in a supply chain or a semantic web application where the application corresponds with Agent 2 in the discussion below and the information accessible in the semantic web by Agent 1.

The discussion of learning an ontology mapping will be based on the two example ontologies given in Section 2 where Agent 1 uses Ontology 1 and Agent 2 uses Ontology 2. Suppose that Agent 1 wishes to know the phone number and email address of a person. Agent 1 knows that the information is (probably) available in an information store managed by Agent 2. Therefore, Agent 1 contacts Agent 2. In order for Agent 1 to put forward its request, the agents first have to establish whether both use the same ontology or whether they use an ontology of which the other agent knows how to map it on its ontology. If the agents use different ontologies and if no mapping is known, the agents should try to establish a mapping.

4.1 Underlying ideas

The way the agents establish a mapping is inspired by *language games* [23, 24]. In a language game, an agent (robot) tries to interpret the utterances of another agent by creating and evaluating associations between the received utterances and categorizations of observed entities, the *joint attention*.

To illustrate the idea behind using language games for ontology mapping, suppose that:

- Agent 1 wishes to communicate about a concept such as a 'person';
- Both agents use different conceptualizations of the concept 'person' (i.e., different ontologies);
- A one-to-one mapping exists between concepts of interest in the two ontologies, that is, the concept 'person' will not be represented by two or more concepts such as 'student' and 'teacher' in another ontology;
- Some instances of the concept 'person' are known by both agents (although it is not known which ones).

Given these assumptions, the agents establish a mapping by the following four steps.

1. Creating a flattened representation of instances of concepts. Such a flattened representation is called an *utterance* and is used in the communication between the two agents.
2. Identifying corresponding instances of concepts in both ontologies by exchanging utterances between the two agents. The pairs utterances representing corresponding instances of concepts, form the *joint attention*.
3. Identifying the corresponding concepts in the two ontologies using the joint attention.
4. Establish a mapping between the corresponding concepts in the two ontologies using the joint attention.

Note that the first step solves the problem of *structural conflicts* by removing the structure of the ontologies while the last two steps handle the problem of *naming conflicts* by determining the most likely mapping between the ontologies, based on the corresponding instances of concepts. Although it should also be possible to learn unit conversions, the handling of data conflicts will not be addressed in this report. In general, the handling data conflicts requires some domain knowledge. In the report, we will only consider mappings that can be learned without any domain knowledge.

4.2 Utterances

To establish a mapping, the agents start with exchanging utterances in order to find instances of concepts that represent the same entity in the agents' environment. Such an utterance represents all relevant information of an instance in a uniform way. Moreover, an utterance also contains an identification of the concept to which the represented instance belongs in order to determine corresponding concepts in the ontologies. Since the two ontologies may have completely different structures (structural heterogeneity) only the values of attributes of a concept are considered. Hence, the aggregation hierarchy of sub-concepts and attributes is flattened. The hierarchy is flattened by representing each attribute by a label followed by the corresponding attribute value. The labels must be unique for every path from the root concept (e.g., the person) to an attribute (e.g., the street name). We cannot use the name of an attribute as a label since it is possible that an attribute is used several times in the definition of a concept. The attribute 'street name', for instance, can be used for the home address and for the working address of a concept 'person'.

By the introduction of labels, an agent transforms a possibly highly structured conceptual hierarchy into a new shallow hierarchy in which the labels represent the new attribute of a concept. It does not matter how agents represent the labels in an utterance, as long as the label is unique. The agent may use, for instance, the term 'pnfn' or a term representing the place of an attribute in the ontology 'person.name.first name' to denote a person's first name in a communication.

An important decision concerns the amount of information to put in an utterance. Since Agent 1 wishes to communicate with Agent 2 about the concept 'person', it must decide which attributes and sub-concepts of the concept 'person' should be included in the utterance. In other words, Agent 1 decides which part of Ontology 1 should be flattened. If, for instance, the concept 'person' in Ontology 1 contains a 'father' and a 'mother' relation with itself and since there are, in principle, no restrictions on the number of ancestors represented in the knowledge base / database described by the ontology, deciding what to include in the utterance is an important issue. Reasons for including, for instance, a person's father, mother, grandfather, grandmother, and so on, are (*i*) Agent 1 may wish to communicate about them and (*ii*) their names are needed to uniquely identify a person.

The following utterances gives an illustration of an utterance representing an instance of the concept 'person' in Ontology 1:

> CONCEPT:person
> person.christian_name:'Archibald'
> person.family_name:'Haddock'
> person.street:'Castle Lane 1'
> person.city:'Marlinspike'
> person.country:'Belgium'
> person.phone number:'06229–421'
> person.email:'haddock@herge.be'
> person.father.christian_name:'Francois'
> person.father.family_name:'de Hadoque'

The value of a label will be represented by a string of characters. This guarantees that there cannot be any confusion about its interpretation; i.e., whether four bytes represent a string of 4 characters or an integer. For numbers the standard translation to strings will be applied. Moreover, Boolean values will be represented by 'true' or 'false'. Since a string may consist of multiple words (e.g., a family name consisting of more than one word or a combination of a street name and a house number), a label's value will be interpreted as a *list of words* separated by punctuation marks.

The following two utterances consisting of label-value pairs for Ontology 1 and Ontology 2 respectively represent instances of the ontologies shown in Figure 1 and Figure 2, respectively.

Ontology 1	**Ontology 2**
CONCEPT:‘person’	CONCEPT:’prsn’
person.christian name:‘Archibald’	pnfn1:‘Archibald’
person.family name:‘Haddock’	pnfn2:‘Haddock’
person.street:‘Castle Lane 1’	pas:‘Castle Lane’
person.city:‘Marlinspike	pan:‘1’
person.country:‘Belgium’	pacn1:‘Marlinspike’
person.phone number:‘06229–421’	pacac:‘06229’
person.email:‘haddock@herge.be’	pacn2:‘Belgium’
	paccc:‘32’
	ppn:‘421’
	pem:‘haddock@herge.be’

4.3 Joint attention

The utterances are used to establish a *joint attention.* A joint attention is a set JA of pairs of utterances (u^1, u^2) where u^1 is an utterance of Ontology 1 and u^2 is an utterance of Ontology 2. Both utterances in a pair (u^1, u^2) *should* represent the same entity in the world.

Agent 2 establishes the joint attention using the utterances u^1 communicated by Agent 1. Given the words in an utterance u^1, Agent 2 searches for a similar instance of a concept of Ontology 2, which will be represented by the utterance u^2. A similar instance denotes the same entity in the world. Agent 2 uses *information retrieval* techniques for unstructured data to determine zero or more matching utterance $u^2_1, \ldots, u^2_r$. To decide whether two utterances u^1 and u^2_i are similar, the probability that u^1 and u^2_i denote the same entity in the world is used. This probability is derived from the corresponding words of the two utterances. Here, a set of *words* consists of the values associated with the labels in an utterance. For example, a person called ‘Haddock’, who lives at ‘Castle’ ‘Lane’ ‘1’ in ‘Marlinspike’, with phone number ‘421’.

We will use the following notations to determine the probability that u^1 and u^2 denote the same entity in the world.

- $U^2 = \{u^2_1, ..., u^2_r\}$ is the set of utterances of instances of Ontology 2 that might denote the same entity as the utterance u^1 of an instance of Ontology 1.
- $id(u^1, u^2_i)$ is a proposition which holds if u^1 and u^2_i denote the same entity in the world.
- e_i denote the words in u^1 that also occur in the utterance u^2_i.
- $E = \langle e_1, ..., e_r\rangle$ is a list containing for each $u^2_i \in U^2$ the set of words e_i in u^1 that also occur in u^2_i. The correspondences between u^1 and each of the utterances in U^2, denoted by E, form the *evidence* for $id(u^1, u^2_i)$.

Given the above listed information, Agent 2 should determine the conditional probability: $P(id(u^1, u^2_i) \mid E)$.

No nesting of concepts In the following discussion of using the probability measure $P(id(u^1, u^2_i) \mid E)$, we assume that concepts of Ontology 2 are independent of each other. At the end of this section, we will relax this assumption.

To determine the conditional probability: $P(id(u^1, u^2_i) \mid E)$, note that the sample space underlying the probability measure consists of instances of mapping problems. Hence, all instances of mapping problems in which the evidence is present should be considered. Given this sample space, we can draw the following three conclusions. The latter two conclusions do not hold in case we consider nested concepts such as the concept ‘person’ and the sub-concept ‘address’ in Ontology 2.

- If $id(u^1, u^2_i)$ holds and if for every label-value pair in the utterance u^1 there is at least one value-label pair in the utterance u^2_i having the same value, then $P(e_i \mid id(u^1, u^2_i)) = 1$. However, u^1 may represent attributes for which there is no corresponding attribute in the corresponding concept of the other ontology. For instance, the concept 'person' in Ontology 2 has no representation of a person's father. Therefore, $P(e_i \mid id(u^1, u_i^2)) \leq 1$. If the number of sets of words that can be denoted by the evidence e_i is sufficiently large, then $P(e_i) \ll P(e_i \mid id(u^1, u_i^2))$. Hence,

$$P(e_i) \ll P(e_i \mid id(u^1, u_i^2)) \leq 1$$

- It is safe to assume for the instances of an ontology that the words occurring in an utterance u_j are independent of the words occurring in u_k with $j \neq k$.
- $P(e_j \mid id(u^1, u_i^2)) \approx P(e_j)$ if $i \neq j$.

Hence,

$$\begin{aligned}
P(id(u^1, u_i^2) \mid E) &= \frac{P(id(u^1, u_i^2)) \cdot P(E \mid id(u^1, u_i^2))}{P(E)} \\
&= \frac{P(id(u^1, u_i^2)) \cdot \prod_{j=1}^{r} P(e_j \mid id(u^1, u_i^2))}{\prod_{j=1}^{r} P(e_j)} \\
&= \frac{P(id(u^1, u_i^2)) \cdot P(e_i \mid id(u^1, u_i^2)) \cdot \prod_{j \neq i} P(e_j \mid id(u^1, u_i^2))}{P(e_i) \cdot \prod_{j \neq i} P(e_j)} \\
&\geq \frac{P(id(u^1, u_i^2)) \cdot P(e_i \mid id(u^1, u_i^2))}{P(e_i)}
\end{aligned}$$

In the last inequation, three terms are important:

- The term $P(e_i)$ denotes the a priori probability that the utterance u_i^2 contains the words e_i from the utterance u^1.
- The term $P(e_i \mid id(u^1, u_i^2))$ denotes the a priori probability that the utterance u_i^2 contains the words e_i from the utterance u^1 if u^1 and u_i^2 represent the same entity in the world.
- The term $P(id(u^1, u_i^2))$ denotes the a priori probability that u^1 and u_i^2 represent the same entity in the world.

The above derived expression for $P(id(u^1, u_i^2) \mid E)$ cannot be used to calculate the probability since Agent 2 does not know $P(e_i \mid id(u^1, u_i^2))$ and $P(id(u^1, u_i^2))$, and since Agent 2 can only estimate $P(e_i)$ if there are no dependencies between the words $w \in e_i$ occurring in the evidence. However, the above derived expression can be used to compare the probabilities $P(id(u^1, u_i^2) \mid E)$ for each $u_i^2 \in U^2$. Suppose $id(u^1, u_i^2)$ holds. Then $P(id(u^1, u_i^2) \mid E)$ is expected to be significantly larger than $P(id(u^1, u_j^2) \mid E)$ if $i \neq j$.

$$\begin{aligned}
P(id(u^1, u_i^2) \mid E) &\gg P(id(u^1, u_j^2) \mid E) \\
\frac{P(id(u^1, u_i^2)) \cdot P(e_i \mid id(u^1, u_i^2))}{P(e_i)} &\gg \frac{P(id(u^1, u_j^2)) \cdot P(e_j \mid id(u^1, u_j^2))}{P(e_j)}
\end{aligned}$$

Since a priori there is no reason to prefer any $id(u^1, u_x^2)$, we assume that $P(id(u^1, u_i^2)) = P(id(u^1, u_j^2))$ for all i and j. Moreover, if $id(u^1, u_i^2)$, then there holds $P(e_i \mid id(u^1, u_i^2)) \geq P(e_j \mid id(u^1, u_j^2))$. Hence,

$$P(e_i) \ll P(e_j)$$

This result implies that Agent 2 should add those pairs to the joint attention that have a significantly lower a priori evidence probability than other pairs.

The probability $P(e_i)$ depends on the probability that a word $w \in e_i$ occurs in the value of a label-value pair in the utterance u_i^2. Let $lv_1, ..., lv_k$ be the label-value pairs in the utterance u_i^2 and let the function $v(\cdot)$ denote the value of a label-value pair. Then, assuming that there are no dependencies between the words occurring in the evidence, Agent 2 can determine $P(e_i)$. $P(e_i)$ is the a priori probability that each word $w \in e_i$ is an element of a value of a label-value pair in u_i^2 while no word $w \in (u^1 \backslash e_i)$ is an element of a value of a label-value pair in u_i^2.

$$\begin{aligned} P(e_i) &\approx P(\bigwedge_{w \in e_i} \exists j : w \in v(lv_j) \wedge \bigwedge_{w \in (u^1 \backslash e_i)} \forall j : w \notin v(lv_j)) \\ &\approx \prod_{w \in e_i} P(\neg \forall j : w \notin v(lv_j)) \cdot \prod_{w \in (u^1 \backslash e_i)} P(\forall j : w \notin v(lv_j)) \end{aligned}$$

In this approximation Agent 2 makes the assumption that the values of label-value pairs are independent of each other. In cases where this assumption is incorrect, the approximated value of $P(e_i)$ will be too low. On the other hand, if a word occurring in the evidence matches with the value of more that one label-value pair, the approximated value of $P(e_i)$ will be too high. Suppose that a word $w \in e_i$ can match with every label-value pair in u_i^2 (which is the case in our experiments). Then the next word $w' \in e_i$ may match with all but one label-value pairs in while $w'' \in e_i$ may match with all but two label-value pairs in u_i^2, and so on. Considering all combinations that can be created in this way cannot be done efficiently.

The probability $P(\forall j : w \notin v(lv_j))$ that a word w of the utterance u^1 occurs in the value of no label-value pair lv_j of u_i^2 is, assuming that independence, the product of the probabilities that the word w does not occur in the value of the label-value pair lv_j.

$$P(\forall j : w \notin lv_j) = \prod_{j=1}^{k} (1 - P(w \in v(lv_j)))$$

Agent 2 can approximate $P(w \in v(lv_j))$ using the instances of Ontology 2 by determining the relative frequency that w occurs in the value of the attribute represented by the label-value pair lv_j. So, Agent 2 can approximate the a priori probability of the evidence $P(e_i)$.

Agent 2 can decide whether $id(u^1, u_i^2)$ holds by evaluating whether $P(e_i) \ll P(e_j)$ holds for every $j \neq i$. Agent 2 can use a threshold value θ^u to make this decision. The threshold value θ^u represents the difference between the estimated a priori probability values. So, Agent 2 will add a pair (u^1, u_i^2) to the joint attention if for every $j \neq i$ there holds:

$$\frac{P(e_j)}{P(e_i)} \geq \theta^u$$

Nested concepts There is one last issue Agent 2 has to take into account when establishing the joint attention JA. In the above outlined approach we assumed that no concept c_a^2 is a sub-concept of a concept c_b^2. This assumption does not hold in general. For instance, in Ontology 2 the concept 'address' is a sub-concept of the concept 'person'.

CONCEPT:prsn	CONCEPT:addr
pnfn1:Archibald	pas:Castle Lane
pnfn2:Haddock	pan:1
pas:Castle Lane	pacn1:Marlinspike
pan:1	pacac:06229
pacn1:Marlinspike	pacn2:Belgium
pacac:06229	
pacn2:Belgium	
paccc:32	
ppn:421	
pem:haddock@herge.be	

Consider an utterance u_i^2 representing an instance of the concept c_a^2. If c_a^2 is a sub-concept of c_b^2, then there exists an utterance u_k^2 representing an instance of c_b^2 such that $e_i \subseteq e_k$. For instance, an utterance describing an instance of the concept 'person' in Ontology 2 contains all the words occurring in an instance of the concept 'address'.

Agent 2 cannot assume that the evidence e_i and e_x is independent in the presence of nested concepts. Therefore, the derivation of the inequation for $P(id(u^1, u_i^2) \mid E)$ must be reconsidered.

Suppose that u^1 and u_i^2 describe the same entity in the world. Then, normally, $e_i = e_k$ will hold, implying:

- $P(e_i) \ll P(e_i \mid id(u^1, u_i^2))$,
- $P(e_i, e_k \mid id(u^1, u_i^2)) = P(e_i \mid id(u^1, u_i^2)) \leq 1$ and
- $P(e_i) = P(e_i, e_k)$.

Hence, the above derived expression for $P(id(u^1, u_i^2) \mid E)$ still holds.

Now suppose that u^1 and u_k^2 describe the same entity in the world. Then $e_i \subset e_k$ will hold, implying:

- $P(e_k) \ll P(e_k \mid id(u^1, u_k^2))$,
- $P(e_i, e_k \mid id(u^1, u_i^2)) = P(e_k \mid id(u^1, u_k^2)) \leq 1$ and
- $P(e_k) = P(e_i, e_k)$.

Hence, the above derived expression for $P(id(u^1, u_i^2) \mid E)$ still holds.

Because the evidences e_i and e_k can be the same or almost the same if c_a^2 is a sub-concept of a concept c_b^2, $\frac{P(e_i)}{P(e_k)} \geq \theta^u$ or $\frac{P(e_k)}{P(e_i)} \geq \theta^u$ need not hold. Therefore,

> if c_a^2 is a sub-concept of a concept c_b^2, Agent 2 should *not* compare $P(e_j)$ with $P(e_k)$ using θ^u.

Agent 2 may consider adding both (u^1, u_i^2) and(u^1, u_k^2) to the joint attention JA. Clearly, at most one pair can be correct. Agent 2 should add (u^1, u_k^2) to the joint attention if there is enough additional evidence for (u^1, u_k^2). That is,

$$\frac{P(e_i)}{P(e_k)} \geq \theta^s$$

for some threshold value θ^s. Otherwise it should add to (u^1, u_i^2) the joint attention.

4.4 Mappings

A mapping M between two ontologies will relate the corresponding concepts in two ontologies and the attributes of these concepts. Ideally, all pairs in the joint attention JA will support the same mapping. However, Agent 2 cannot prevent that a pair of utterances is incorrectly added to JA because of coincidental correspondences. Hence, Agent 2 must determine the mapping for which it has the highest support. A second reason for determining the mapping with the highest support is because a pair of utterances may support more than one mapping. The latter can occur if two label-value pairs of an utterance have the same value.

To determine the mapping with the highest support, Agent 2 might start with determining all mappings that are supported by at least one pair of utterances in the joint attention JA. Let $M_1, \ldots, M_m$ be all these mappings and let JA_i denote the subset of JA supporting the mapping M_i. Agent 2 preferably should determine the conditional probability $P(M_i \mid \mathbf{E})$ where $\mathbf{E} = \{E_k \mid (u_k^1, u_k^2) \in JA\}$ denotes all the evidence for the joint attention JA.

Suppose that a mapping M_i is incorrect, denoted by $\neg M_i$. Then, all pairs $(u^1, u^2) \in JA_i$ are incorrectly added to JA, denoted by $\neg JA_i$. Hence,

$$\begin{aligned} P(\neg M_i \mid \mathbf{E}) &= P(\neg JA_i \mid \mathbf{E}) \\ &= \prod_{id(u^1,u^2) \in JA_i} P(\neg id(u^1, u^2) \mid \mathbf{E}) \\ &\approx \alpha_i^{|JA_i|} \end{aligned}$$

where $\alpha_i \approx P(\neg id(u^1, u^2) \mid \mathbf{E})$ is assumed to hold for all $id(u^1, u^2) \in JA_i$. Unfortunately, Agent 2 does not have enough information to estimate a value for α_i. Hence, based on the above result, Agent 2 can only say that the probability that the mapping M is incorrect, is exponentially decreasing in the size of JA_i.

Assuming that α_i is more or less the same for all mappings M_i, i.e., $\alpha \approx \alpha_i$ for every i, and assuming $P(M \mid \mathbf{E}) = 0$ for every mapping M not belonging to $M_1, \ldots, M_m$, an estimation of the odds that a mapping is correct can be made.

$$\begin{aligned} O(M_i \mid \mathbf{E}) &= \frac{P(M_i \mid \mathbf{E})}{P(\neg M_i \mid \mathbf{E})} \\ &= \frac{P(\bigwedge_{j \neq i} \neg M_j \mid \mathbf{E})}{P(\neg M_i \mid \mathbf{E})} \\ &\approx \frac{\prod_{j \neq i} \alpha_j^{|JA_j|}}{\alpha_i^{|JA_i|}} \\ &\approx \alpha^{\sum_{j \neq i} |JA_j| - |JA_i|} \end{aligned}$$

Since $\alpha < 1$, this result implies that:

$$\frac{|JA|}{\sum_{j \neq i} |JA_j|} \geq \theta^m > 1$$

should hold for some threshold θ^m. Since the sets JA_j are disjoint, $\sum_{j \neq i} |JA_j| = |JA - JA_i|$. Hence, we can write $|JA_i| \geq \theta^m \cdot |JA_i^-|$ with $JA_i^- = JA - JA_i$.

The above result also provides Agent 2 with a way to create and evaluate a mapping step by step. Since $|JA_i| \geq \theta^m \cdot |JA_i^-|$ must hold for a mapping M_i, it must also hold for each element making up the mapping. Therefore, Agent 2 can derive a mapping by first identifying the corresponding concept, and subsequently, association between the labels.

Corresponding concepts A pair (u^1, u^2) in the joint attention comprises an instance of a concept c^1 of Ontology 1 and an instance of a concept c^2 of Ontology 2. Since the instances are probably describing the same entity in the world, the concept c^1 and c^2 should also correspond assuming that a one to one mapping exists between corresponding concepts in both ontologies. For instance, the concept 'person' in one ontology will *not* be represented by the concepts 'student' and 'teacher' in the other ontology.

Since there is no absolute certainty that the pairs of utterances that are added to the joint attention are correct, it is possible that concept c^1 is identical to the concept c_i^2 (i.e., $id(c^1, c_i^2)$), according to one pair in the joint attention, and to the concept c_j^2 according to another pair. Clearly, assuming a one to one mapping, one of the two must be incorrect. If JA_i denotes the pairs of utterances supporting $id(c^1, c_i^2)$ and if JA_i^- denotes the pairs of utterances supporting other correspondences between concepts: $id(c^1, c_j^2)$ with $j \neq i$, then the result of this subsection implies that $id(c^1, c_i^2)$ may hold according to a mapping M_i if $|JA_i| \geq \theta^m \cdot |JA_i^-|$.

After having determined all pairs (c_i^1, c_i^2) of corresponding concepts; i.e., all pairs for which $id(c_i^1, c_i^2)$ holds, Agent 2 can remove those pairs (u_j^1, u_j^2) from the joint attention for which $c_i^1 = c(u_j^1)$ and $c_i^2 \neq c(u_j^2)$, where $c(u_x^y)$ denotes the concept of which utterance u_x^y is an instance. This gives us the following *pruned* joint attention:

$$\widehat{JA} = \{(u_j^1, u_j^2) \in JA \mid c_i^1 = c(u_j^1) \Rightarrow c_i^2 = c(u_j^2)\}$$

If the pruned joint attention is empty; i.e., if no pairs pass the threshold θ^u or no corresponding concepts are identified given θ^m, then no associations can be created. This is favored over finding an ontology mapping that is probably not correct.

Creating associations After establishing the corresponding concepts, Agent 2 will try to establish a mapping between the attributes that make up the concept c^1 respectively c^2. To establish a mapping Agent 2 uses pairs of utterances (u^1, u^2) from the pruned joint attention $\widehat{JA}$.

Given a pair of utterances from the joint attention, Agent 2 proposes associations between the labels of two utterances based on the proportion of corresponding words. An association consists of an Agent 1 label and an ordered list of operations on Agent 2 labels. Possible operations are:

label(x): denotes the value in the utterance denoted by the label with name x.

first(x, s): as label(x), but returns the value up to the first occurrence of separator s. Possible separators are: ‘ ’, ‘,’, ‘;’, and TC (a type change, i.e., a transition from numbers to letter or vice versa).

last(x, s): as first(x, s), but returns the value starting after the first occurrence of separator s.

conc($t, x_1, \ldots, x_n$): concatenates all the elements $x_1, \ldots, x_n$ and inserts t as separator.

null: does nothing; used when no operation is applicable.

The mapping operations were chosen on the one hand to be expressive enough to enable useful mappings, and on the other hand to be limited in order to prevent a combinatorial explosion.

The following example illustrates a mapping from Ontology 1 to Ontology 2.

label(pan) $\leftarrow$ first(label(person.street),TC)

Agent 2 searches through a space of possible associations guided by the proportion of words that instances of concepts have in common. Agent 2 accepts an association *assoc* as being correct if it is supported by enough pairs in the pruned joint attention $\widehat{JA}$. That is, if $\widehat{JA}_i$ denotes the pairs of utterance in $\widehat{JA}$ supporting the association $assoc_i$ and if $\widehat{JA}^-$ denotes the pairs of utterance in $\widehat{JA}$ supporting other associations $assoc_j$ with $i \neq j$, then the result of this subsection implies that $assoc_i$ may hold according to a mapping M_k if $|\widehat{JA}_i| \geq \theta^m \cdot |\widehat{JA}_i^-|$.

In order to get the best possible result, the first association ‘label(x) $\leftarrow \ldots$’ for a label-value pair ‘label(x)’ that is added to a mapping M_k, is the one for which $\frac{\widehat{JA}_i}{\widehat{JA}_i^-}$ is maximal. Subsequently, Agent 2 prunes the joint attention using the association ‘label(x) $\leftarrow \ldots$’ creating a new pruned joint attention $\widehat{\widehat{JA}}$. Next, Agent 2 repeats the process by searches for the next best association using the new joint attention $\widehat{\widehat{JA}}$. Agent 2 continues searching for associations and pruning the joint attention till no more associations can be found that are supported by enough pairs of utterances in the joint attention.

5 Experiments with basic ontology mappings

We have evaluated our method for learning an ontology mapping through two series of experiments. In the experiments we have investigated the number of errors that where made in establishing the joint attention, in determining the corresponding concept, and in creating a mapping. These aspects depend, of course, on the threshold values θ^u, θ^s and θ^m. A number of other factors also influence the success of learning a mapping.

- Increasing the number of labels in an utterance makes the mapping problem easier (because it becomes more clear that two utterances denote the same entity).
- Increasing the number of words in the set W from which the values of an attribute are chosen makes the mapping problem easier (because there is less room for confusion).
- The occurrence of sub-concepts and super-concepts makes the mapping problem harder (because they overlap, and especially if they differ on only a few labels in an utterance).
- Splitting and concatenating label values makes the mapping problem harder (because the search space becomes larger).
- Labels in one ontology that do not occur in the other ontology make the mapping problem harder (because there is more room for confusion).
- Mistakes or differences in utterances makes the mapping problem harder (because it makes the matching utterances less similar).

We investigated the influence of the above mentioned aspects on the success of learning a mapping in two experiments.

The goal of the first experiment was to investigate how well our approach can find matching utterances and the correct associations. Therefore, we assumed one concept in both ontologies. To complicate the determination of the associations, we introduced one label of which the value had to be split in two association and two labels that had to be concatenated in an association. To make finding corresponding utterances hard, the maximum number of words in a label was set to 2.

We performed the experiments for different values for θ^m, for different numbers of unrelated labels, and for different numbers of words in W. We also experimented with the influence of noise in the correct joint attention.

The goal of the second experiment was to investigate how well our approach can distinguish different yet similar concepts.

Experiment 1 In this series of experiments, Ontology 1 consisted of one concept c^1 which had to be mapped to a concept c^2 of Ontology 2. Moreover, if ℓ^i_j denotes the j-th label in an utterance of Ontology i, then:

- label ℓ^1_1 in u^1 corresponded with label ℓ^2_1 in u^2,
- label ℓ^1_2 in u^1 corresponded with labels ℓ^2_2 and ℓ^2_3 in u^2,
- label ℓ^2_4 in u^2 corresponded with labels ℓ^1_3 and ℓ^1_4 in u^1,
- no label in u^2 corresponded with in ℓ^1_5 and ℓ^1_6 in u^1.
- no label in u^1 corresponded with in ℓ^1_5 and ℓ^1_6 in u^2.

Ontologies 1 and 2 were randomly generated for each experiment, making sure that 10 instances of c^1 corresponded with 10 instances of c^2. In both ontologies, each value of a label consists of 2 words, with the exception of ℓ^1_3, ℓ^1_4, ℓ^2_2 and ℓ^2_3 since ℓ^1_3 and ℓ^1_4 corresponded with ℓ^2_4, and since ℓ^2_2 and ℓ^2_3 corresponded with ℓ^1_2. The total number of instances of each ontology was 1000. Given these ontologies, the agents established a mapping between them. In the experiments the following values for θ^u, θ^m, and the size of the set of words W were chosen: $\theta^u \in \{1000\}$, $\theta^m \in \{1, 1.5, 2, 3\}$ and $|W| \in \{100, 250, 500, 1000\}$.

In each experiment, we determined the recall and the precision for the joint attention, and counted the number of correct associations in a mapping. Also the number of times that a mapping was not possible was determined. If in a run no mapping was possible, the run was ignored when computing the other three variables. Table 1 shows some of the average results over 100 runs (for each run new random instances were created). For $W = 100$, often it is not possible to find a mapping (0.28-0.40). If nevertheless a mapping is found, the result is acceptable. For increasing $|W|$, the precision increases and consequently also the correct associations. The high precision for $|W| = 100$ is somewhat misleading, since in most runs there was no precision. The best results were found for $\theta^m = 1.0$; for each run the correct mapping was found.

One might expect that increasing the value of θ^m would increase the number of correct mappings found instead of decreasing this number as we see in Table 1. Increasing θ^m implies that we need a larger joint attention. Table 2 shows that increasing the number of instances represented in both ontologies from 10 to 30 improves the result.

Next, we increased the number of labels n that had no correspondence, making it more difficult to find a mapping. The number of instances represented in both ontologies was set back to 10. Table 3 shows the results. Not surprisingly, the results are worse for increasing n. The effect is most notable for $W = 100$ (almost a factor 3 when n increases from 2 to 4), but decreases for increasing W. Note that for $n = 4$, the ontologies have more non-corresponding labels than corresponding labels.

Finally, we investigated the consequences of adding noise to the instances. For the instances of Agent 1 that were part of the correct joint attention, a percentage of the words was replaced by randomly chosen other words. Consequently, labels that were supposed to match did not match anymore for all 10 joint attention pairs. Table 4 shows the results. With the increase of noise,

	θ^m	100	250	500	1000
Recall	1.0	0.162	0.981	1.000	0.998
Precision	1.0	0.828	0.846	0.994	1.000
No mapping	1.0	0.35	0.00	0.00	0.00
Associations	1.0	0.839	0.847	0.993	1.000
Recall	1.5	0.175	0.971	0.997	1.000
Precision	1.5	0.823	0.664	0.941	0.997
No mapping	1.5	0.32	0.00	0.00	0.00
Associations	1.5	0.833	0.613	0.920	0.967
Recall	2.0	0.168	0.979	1.000	1.000
Precision	2.0	0.817	0.591	0.942	0.947
No mapping	2.0	0.28	0.00	0.00	0.00
Associations	2.0	0.817	0.480	0.920	0.920
Recall	3.0	0.160	0.975	0.999	1.000
Precision	3.0	0.778	0.516	0.821	0.821
No mapping	3.0	0.40	0.00	0.00	0.00
Associations	3.0	0.744	0.360	0.680	0.667

Table 1: Results of Experiment 1, for various values of $|W|$ and θ^m.

	θ^m	100	250	500	1000
Recall	3.0	0.171	0.970	1.000	1.000
Precision	3.0	0.952	0.958	1.000	1.000
No mapping	3.0	0.3	0.00	0.00	0.00
Associations	3.0	0.929	0.933	1.000	1.000

Table 2: Results of Experiment 1, for 30 corresponding instances.

recall becomes slightly worse, and precision and associations are worse. The differences between 5% and 10% are greater than the differences between 0% and 5%.

Experiment 2 In this series of experiments, Ontology 1 consisted of one concept c^1 which had to be mapped to a concept c^2_2 of Ontology 2. Ontology 2 consisted of four concepts in total: c^2_1, c^2_2, c^2_3 and c^2_4. The concept c^2_3 was a sub-concept of c^2_2 and c^2_2 was a sub-concept of c^2_1. Concept c^2_4 was unrelated to the other concepts. The instances of concepts c^2_1 and c^2_4 each consisted of five single valued labels. The instances of concept c^2_2 contained n additional single valued labels with respect to the corresponding instances of c^2_1 and the instances of concept c^2_3 contained n additional single valued labels with respect to the corresponding instances of c^2_2. Instances of the concept c^1 contained $n + 5$ single valued labels.

Ontology 1 and 2 were randomly generated for each experiment, making sure that 10 instances

	n	100	250	500	1000
Recall	3	0.040	0.887	0.990	1.000
Precision	3	0.280	0.707	0.855	1.000
No mapping	3	0.53	0.00	0.00	0.00
Associations	3	0.436	0.727	0.853	1.000
Recall	4	0.025	0.270	0.980	1.000
Precision	4	0.135	0.813	0.659	0.971
No mapping	4	0.49	0.26	0.00	0.00
Associations	4	0.327	0.847	0.640	0.967

Table 3: Results of Experiment 1, for various numbers of unrelated labels (with $\theta^m = 1.0$) In Table 1: $n = 2$.

	Noise (%)	100	250	500	1000
Recall	0	0.162	0.981	1.000	0.998
Precision	0	0.828	0.846	0.994	1.000
No mapping	0	0.35	0.00	0.00	0.00
Associations	0	0.839	0.847	0.993	1.000
Recall	5	0.161	0.954	0.998	1.000
Precision	5	0.781	0.540	0.744	0.741
No mapping	5	0.29	0.00	0.00	0.00
Associations	5	0.836	0.742	0.967	0.977
Recall	10	0.126	0.962	0.996	0.998
Precision	10	0.699	0.518	0.744	0.751
No mapping	10	0.31	0.00	0.00	0.00
Associations	10	0.712	0.597	0.937	0.953

Table 4: Results of Experiment 1 with noise ($\theta^m = 1$, $n = 2$).

	n	100	250	500	1000
Recall	1	0.788	0.980	0.989	0.998
Precision	1	1.000	1.000	1.000	1.000
Concept	1	1.000	1.000	1.000	1.000
Associations	1	1.000	1.000	1.000	1.000
Recall	5	0.061	1.000	1.000	1.000
Precision	5	1.000	1.000	1.000	1.000
Concept	5	1.000	1.000	1.000	1.000
Associations	5	0.889	1.000	1.000	1.000

Table 5: Results of Experiment 2 for $\theta^m = 1.5$, $s = 5$.

of c^1 corresponded with 10 instances of c_2^2. Ontology 1 contained 1000 randomly generated instances of the concept c^1 and Ontology 2 contained 990 randomly generated instances of the concept c_1^2 and an additional 10 instances of c_1^2 where generated in such a way that the corresponding instances of c_2^2 correspond with 10 instances of c^1. Moreover, Ontology 2 contained 1000 randomly generated instances of concept c_4^2. Given these ontologies, the agents established a mapping between them. In the experiments θ^s was set to 5, θ^m was set to 1.5, and $|W| \in \{100, 250, 500, 1000\}$. The results of these experiments are shown in Table 4. Note that the value $\theta^s = 5$ is sufficiently low to guarantee that one additional corresponding label-value pair was sufficient to exceed the threshold value for all values of $|W|$.

Table 5 shows that in all cases only correct mappings are found. Except for $W = 100$ and $n = 5$, the correct mappings are always identified.

6 Extensions

The experiments have shown that the proposed approach is highly successful in learning a mapping if a few entities in the world are represented in both ontologies. In this section we discuss two extensions that will make the proposed approach more generally applicable.

Context dependent mappings The experiments show that agents can successfully learn a mapping between their ontologies. There are, however, a number of restrictions on mappings that can be learned by the agents, such as learning context dependent mappings. A context dependent mapping is a mapping in which the correct association depends on the values of certain attributes. For instance, the way in which the street name and the house number are represented in a single attribute depends on the country. In the Netherlands the house number is placed behind the street name while in the USA it is in front of the street name.

An agent can, in principle, learn a context dependent mapping by generating an association between attributes as it did before. For each association the agent proposes, it also formulates a context. A context is a set of label-value pairs for which the association is correct. To illustrate why a context might be a set of label-value pairs, suppose that the way a street name and a house number is combined differs from city to city in some countries (e.g., Belgium). Then the context must contain the name of the country as well as the name of the city.

An association may be valid in more than one context. Germany and The Netherlands, for instance, combine the street name and house number in the same way, resulting in two different contexts; one in which the attribute country has the value 'Germany' and one in which it has the value 'The Netherlands'.

Multiple contexts in combination with contexts that consist of a set of label-value pairs significantly complicate the learning process. An agent could of course use all label-value pairs as a context. This results in context for an association that consists of a complete utterance. Such an approach does not generalize to utterances that do not occur in the joint attention. Hence, a context should contain no more information than necessary. Unfortunately, determining a minimal context is an NP-Hard problem. We can prove this by reducing the Minimal Disjunctive Normal Form problem to a minimal context determination problem.

A Minimum Disjunctive Normal Form problem is a problem in which we have a set of Boolean variables U, and a set A of truth value assignments to these variables. The question to be answered is whether there exists a formula over U in *disjunctive normal form* with no more than K disjuncts that is true for every assignment in A but for no assignment not in A.

Proposition 1 *Learning a context dependent mapping of which the context is minimal is an NP-hard problem.*

Proof We reduce the problem by representing each Boolean variable in U as an attribute and a truth value assignment to the variable in U as a label-value pair. Moreover, we can interpret each assignment in A as a description of all instances for which an association is correct, all assignments not in A as a description of all instance for which an association is incorrect. By determining all minimal contexts, we can check whether there exists a formula in disjunctive normal form with no more than K disjuncts. For, each minimal context corresponds with a disjunct. $\square$

Though learning a context dependent mapping is, in general, infeasible, there are special cases where a context dependent mapping is feasible. Suppose, for instance, that a concept 'person' in one ontology is represented by two concepts, 'student' and 'teacher', in another ontology. If the former ontology contains an attribute indicating whether the person is a student or a teacher, a context dependent mapping between concepts can be learned efficiently. This is possible because there is only one attribute value for each mapping. Moreover, instead of a single attribute, also sets of attributes can form a context that can be learned efficiently, provided that (i) for each mapping there is exactly one context of attributes, and (ii) there are enough pairs in the joint attention to exclude irrelevant attributes from the context.

Mappings between sets of concepts Until now we have assumed that Agent 1 wishes to communicate about one concept and that Agent 2 tries to establish a mapping between this concept and a corresponding concept in its ontology. If Agent 1 wishes to communicate about more than one concept, the same approach can be used if the concepts are unrelated. Usually, however, there will exist relations between the concepts Agent 1 wishes to communicate about. When establishing a mapping, the relations between the concepts must also be learned.

Suppose that Agent 1 wishes to communicate about the concepts $c^1_1, \ldots, c^1_k$. Hence, Agent 1 has to formulate utterances for each of the concepts. If there are relations between two of these concepts: c^1_i and c^1_j, Agent 1 could, of course, include c^1_j in the utterance of an instance of c^1_i. This will make it hard for Agent 2 to learn that there is a relation between c^1_i and c^2_j. Therefore, in an utterance describing an instance of concept c^1_i, labels denoting a relation with other concepts c^1_j should be added.

CONCEPT:person
person.christian_name:'Archibald'
person.family_name:'Haddock'
person.street:'Castle Lane 1'
person.city:'Marlinspike'
person.country:'Belgium'
person.phone number:'06229–421'
person.email:'haddock@herge.be'
person.father⟶CONCEPT:'person'
person.mother⟶CONCEPT:'person'

The labels denoting a relation between c^1_i and c^1_j; e.g., the last two labels in the utterance above, will not be used by Agent 2 to establish a mapping between concept c^1_i and a concept c^2_l of its ontology. Hence, after ignoring these labels, what remains is the problem of establishing a mapping between a concept of Ontology 1 and a concept of Ontology 2 as has been discussed in the preceding subsections. Having established a mapping for each concept c^1_i of Ontology 1 with a concept c^2_l of Ontology 2, the mappings have to be extended in order to incorporate the labels such as:

person.father⟶CONCEPT:'person'
person.mother⟶CONCEPT:'person'

that have been ignored so far. Since Agent 2 has established which concept in its ontology corresponds with which concept communicated by Agent 1, given a pair of utterances (u^1, u^2), Agent 2 can determine the labels in an utterance u^2 that refer to the same concepts as labels in the utterance u^1. Next Agent 2 has to determine whether instances of the concepts referred to are the same. This is especially important if there is more than one reference to the same concept, as is the case in the above example. For each pair of utterance in the joint attention and for each label referring to a concept, Agent 2 has to request Agent 1 to send the utterance v^1 describing the actual instance of the concept referred to. Subsequently Agent 2 has to determine whether it has a corresponding instance of the corresponding concept; i.e., Agent 2 determines pairs of utterances (v^1, v^2) forming a joint attention for the references. Using the new joint attention, Agent 2 can evaluate associations between labels representing references in the same way as it evaluates other associations between labels.

7 Conclusions

We have proposed a successful approach for learning mappings between two ontologies. The learning method is based on exchanging utterances representing instances of concepts. If a small number (10) of instances of concepts are represented using both ontologies, then a 100% correct mapping can be learned. The method can cope with noise in the data and also can indicate when a mapping is not possible.

The strongest assumption of the proposed approach is the requirement that a small number of instances of corresponding concepts are represented in both ontologies. Therefore future research will focus on relaxing this requirement.

In the paper we assumed that data conflicts do not occur since handling data conflicts often requires domain specific knowledge. Initial experiments [11] have shown that some data conflicts caused by the use of different units or different precision can be learned. Further research is, however, required.

References

[1] S. Bergamaschi, S. Castano, S. Vermercati, S. Montanari, and M. Vincini. An intelligent approach to information integration. In *International Conference on Formal Ontology in Information Systems (FOIS'98)*, 1998.

[2] M. Burnstein, D. McDermott, D. R. Smith, and S. J. Westfold. Semantic integration of semistructured and structured data sources. *Autonomous Agents and Multi-Agent Systems*, 6:265–286, 2003.

[3] A. Doan, J. Madhavan, P. Domingos, and A. Halevy. Learning to map between ontologies on the semantic web. In *The Eleventh International WWW Conference*, 2002.

[4] A. Farquhar, R. Fikes, and J. Rice. The ontolingua server: a tool for collaborative ontology construction. In *Proceedings of the Tenth Knowledge Acquisition for Knowledge-Based Systems Workshop*, 1996.

[5] A. Firat, S. Madnick, and B. Grosof. Knowledge intergration to overcome ontological heterogeneity: challenges from financial information systems. In *Twenty-Third International Conference on Information Systems*, 2002.

[6] M.P. Gallaher, A.C. O'Connor, Jr. J.L. Dettbarn, and L. T. Gilday. Cost analysis of inadequate interoperability in the u.s. capital facilities industry. Technical Report NIST GCR 04-867, National Institute of Standards and Technology, 2004.

[7] T. R. Gruber. Ontolingua: A mechanism to support portable ontologies. Technical Report KSL 91-66, Stanford University, Knowledge Systems Laboratory, 1992.

[8] T. R. Gruber. Toward principles for the design of ontologies used for knowledge sharing. *International Journal of Human-Computer Studies*, 43:907–928, 1995.

[9] P. Haase and B. Motik. A mapping system for the integration of owl-dl ontologies. In *Proceedings of the first international workshop on Interoperability Of Heterogeneous Information Systems (IHIS'05)*, pages 9–16, 2005.

[10] J. Hammer and D. McLeod. An approach to resolving semantic heterogeneity in a federation of autonomous, heterogeneous database systems. *Journal for Intelligent and Cooperative Information Systems*, 2(1):51–83, 1993.

[11] D. Jehoul. Interoperabiliteit. Master Thesis CS-05-20, Universiteit Maastricht, 2005.

[12] W. Kim and J. Seo. Classifying schematic and data heterogeneity in multidatabase systems. *IEEE Computer*, 24(12):12–18, 1991.

[13] S. Lacher and G. Groh. Facilitating the exchange of explixit knowledge through ontology mappings. In *14th International FLAIRS conference*, 2001.

[14] Y. Lee, K. Supekar, and J. Geller. Ontology integration: Experience with medical terminologies. *Computers in Biology and Medicine*, 36:893–919, 2006.

[15] D. L. McGuinness and F. van Harmelen. Owl web ontology language overview. *W3C Recommendation 10 February 2004*, 2004.

[16] Tova Milo and Sagit Zohar. Using schema matching to simplify heterogeneous data translation. In *Proc. 24th Int. Conf. Very Large Data Bases, VLDB*, pages 122–133, 1998.

[17] M. P. Papazoglou, N. Russell, and D. Edmond. A translation protocol achieving consensus of semantics between cooperating heterogeneous database systems. In *Conference on Cooperative Information Systems*, pages 78–89, 1996.

[18] D. Pérez-Rey, V. Maojo, M. García-Remesal, R. Alonso-Calvo, H. Billhardt, F. Martin-Sánchez, and A. Sousa. Ontofusion: Ontology-based integration of genomic and clinical databases. *Computers in Biology and Medicine*, 36:712–730, 2006.

[19] H. S. Pinto. Some issues on ontology integration. In *IJCAI-99 workshop on Ontologies and Problem-Solving Methods (KRR5)*, 1999.

[20] S. Prasad, Y. Peng, and T. Finin. Using explicit information to map between two ontologies. In S. Cranefield, T. Finin, and S. Willmott, editors, *Ontologies in Agent Systems workshop*, 2002.

[21] A. P. Sheth and J. A. Larson. Federated database systems for managing dristributed, heterogeneous, and autonomous databases. *ACM Computing Serveys*, 22:183–236, 1990.

[22] L.-K. Soh. Multiagent distributed ontology learning. In S. Cranefield, T. Finin, and S. Willmott, editors, *Ontologies in Agent Systems workshop*, 2002.

[23] L. Steels. Emergent adaptive lexicons. In P. Maes, editor, *From Animals to Animats 4: Proceedings of the Fourth International Conference On Simulating Adaptive Behavior*, Cambridge Ma., 1996. The MIT Press.

[24] L. Steels and P. Vogt. Grounding adaptive language games in robotic agents. In C. Husbands and I. Harvey, editors, *Proceedings of the Fourth European Conference on Artificial Life*, Cambridge Ma. and London, 1997. MIT Press.

[25] G. Stumme and A. Maedche. Ontology merging for federated ontologies on the semantic web. In *Proceedings of the International Workshop for Foundations of Models for Information Integration (FMII-2001)*, 2001.

[26] R. M. van Eijk, F. S. de Boer, W. van der Hoek, and J.J. Ch. Meyer. On dynamically generated translators in agent communication. *International Journal of Intelligent Systems*, 16:587–607, 2001.

[27] P. R. S. Visser, D. M. Jones, T. J. M. Bench-Capon, and M. J. R. Shave. An analysis of ontological mismatches: Heterogeneity versus interoperability. In *AAAI 1997 Spring Symposium on Ontological Engineering*, Stanford, USA, 1997.

[28] H. Wache, T. Vögele, U. Visser, H. Stuckenschmidt, G. Schuster, H. Neumann, and S. Hübner. Ontology-based integration of information - a survey of existing approaches. In *IJCAI 2001 Workshop on Ontologies and Information Sharing*, 2001.

[29] S. Weibel, J. Kunze, C. Lagoze, and M. Wolf. Dublin core metadata for resource discovery. *The Internet Society*, RFC 2413, 1998.

[30] P. C. Weinstein and W. P. Birmingham. Agent communication with differentiated ontologies: eight new measures of description campatibility. Technical Report CSE-TR-383-99, Artificial Intelligence Laboratory, University of Michigan, 1999.

[31] F.J. Wiesman and N. Roos. Domain independent learning of ontology mappings. In *AAMAS2004*, pages 844–851, 2004.

[32] F.J. Wiesman, N. Roos, and P. Vogt. Automatic ontology mapping for agent communication. In *BNAIC*, pages 291–298, 2001.