Domain independent learning of ontology mappings

Floris Wiesman
Department of Medical Informatics
Universiteit van Amsterdam
P.O.Box 22700, 1100 DE Amsterdam
f.j.wiesman@amc.uva.nl

Nico Roos
IKAT
Universiteit Maastricht
P.O.Box 616, 6200 MD Maastricht
roos@cs.unimaas.nl

Abstract

This paper proposes a domain independent method for handling interoperability problems by learning a mapping between ontologies. 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 denoting the same entity in the world using information retrieval techniques, followed by proposing and evaluating mappings between the ontologies using the pairs of instances. For each step of this method, the likelihood that a decision is correct is taken into account. 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.

1. Introduction

The rapid growth of networks such as the Internet offers new possibilities for accessing information. At the same time increasingly more information is generated. To keep up with this growing supply of information, intelligent tools are required. Agent technology is one of the most promising ways of handling the growing supply of information. The reason for this is that communication and collaboration are central issues of Multi Agent Systems (MAS).

Agent communication languages such as ACL and KQML provide a standard for agent communication in an open MAS. These languages enable an agent to specify the intention and the content of a message as well as the protocol, the language, and the ontology that are used. For the protocol and the language, some standards are available and should be known by the communicating agents (e.g., FIPA protocols, KIF, and SL).

The ontology [5] used in a communication depends on the subject of the communication. Standard ontologies such as the Dublin Core and the ontologies of the Ontolinga library, and languages for specifying ontologies such as DAML+OIL and OWL, are currently being developed. Nevertheless, since the number of possible subjects is almost infinite and since the concepts used for a subject can be described by different ontologies, the development of generally accepted standards will take a long time. This lack of standardization, which hampers communication and collaboration between agents, is known as the *interoperability problem* [11, 20, 21].

The interoperability problem also occurs in the area of heterogeneous databases [1, 6, 9, 10]. The Internet makes it possible to access (legacy) databases that have been developed in isolation, either because they belong to different legal entities or because they are located at different sites between which no communication was possible before the era of the Internet. Performing queries that require access to several of these databases is impossible unless we know how to relate the information of the databases. One way to relate the information of different databases is to use an ontology to describe the underlying semantic structure of a database.

This paper proposes a method that is based on exchanging instances of concepts that are defined in the ontologies. Before presenting our method for learning an ontology mapping, in Section 2 we 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 concludes the paper.

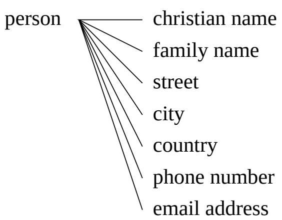
2. Interoperability

In order to reach interoperability, two problems must be dealt with, namely: *structural heterogeneity* and *semantic heterogeneity* [13, 20]. 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. In a multi agent system an ontology is the basis for communication. The actual way information is stored by an agent is shielded from the environment by the agent.

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 difference between ontologies[1]: (1) different semantic structures, *structural conflicts* [1], (2) different names for the same type of information or the same name for (slightly) different types of information, *naming conflicts* [1, 20], and (3) different representations of the same data, *data conflicts* [7]. The data conflict 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).

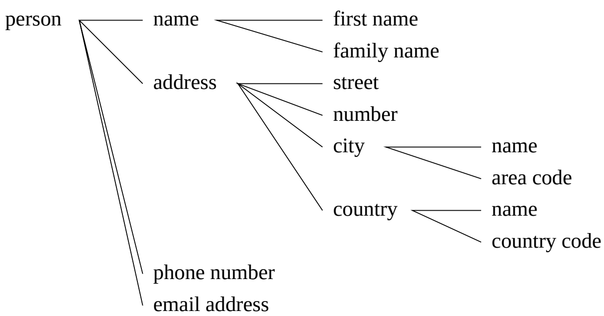
The following two ontologies illustrate some forms of semantic heterogeneity. Both ontologies define a concept 'persons' in terms of relations, sub-concepts and attributes. In these simple ontologies, the attributes are located on the left hand side and the lines represent 'has_a'-relations. The approach proposed in this paper in not limited to ontologies with only 'has_a'-relations. Other type of relations, such as 'father' and 'mother', may also be used.

Ontology 1



In ontology 1, 'street' also describes the house number and 'phone number' describes the country code, the area code, and the local number.

Ontology 2



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.

1 A slightly different categorization of differences between ontologies is given in [19]

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 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 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'.

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.

3. Shortcomings in current approaches

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, 6, 9, 11, 21]. 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.

Besides the problem that the above mentioned assumptions often cannot be met, deriving a common ontology is an indirect way of establishing a mapping between two ontologies.

Some approaches address the problem of establishing a mapping directly [2, 3, 4, 8, 10, 14, 17, 18]. Firat et al. [4] 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. [10] 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. [18] give a characterization of establishing a mapping using first order logic. 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.

In [22], the authors have outlined the initial ideas behind their learning method. Prasad et al. [12] have adapted this method for learning a mapping between ontologies for 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, 8, 14]. Doan et al. [3] (GLUE) and Lacher & Groh [8] use the classifiers to estimate the (joint) probability that 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 [14] are concept classification rules. Concept translations are derived by matching the classification rules.

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 indirect, 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'.

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 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 [15, 16]. 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';
- that 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.

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. Determining corresponding instances of concepts in both ontologies by exchanging utterances between the two agents. The pairs of corresponding instances that are identified, form the *joint attention*.
3. Determining 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 paper.

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.has.name.has.first name' to denote a person's first name in a communication.

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 there is a standard translation to strings which will be applied. Moreover, Boolean values will be represented by 'true' or 'false'. Since a string may consist of different 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 examples represent an utterances consisting of label-value pairs for Ontology 1 and Ontology 2 respectively; see Section 2.

Ontology 1

CONCEPT:person
person.has.christian name:Archibald
person.has.family name:Haddock
person.has.street:Castle Lane 1
person.has.city:Marlinspike
person.has.country:Belgium
person.has.phone number:06229–421
person.has.email address:haddock@herge.be

Ontology 2

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

An important issue is deciding how much 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 in an important issue. Reasons for including, for instance, a person's father, mother, grandfather, grandmother, and so on, are (*i*) because Agent 1 wishes to communicate about them and (*ii*) their names are needed to uniquely identify a person.

After receiving an utterance u^1 from Agent 1, Agent 2 flattens the part of its ontology that leads to the best matching utterance u^2. Agent 2 uses *information retrieval* techniques for unstructured data to determine the best matching utterance u^2. That is, Agent 2 searches for an instance of a concept and determines which sub-concepts provide relevant (i.e., matching) information.

4.3. Joint attention

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 similarity measure based on probabilities is used in the search process. This measure indicates the odds that the two instances represented by their utterances denote the same entity in the world given 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'.

The similarity measure is defined using the conditional probability that two utterance are representing the same entity in the world given the corresponding words in the two utterances. Let $U^2 = \{u_1^2, ..., u_r^2\}$ be 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. Moreover, let $id(u^1, u_i^2)$ hold if the utterances u^1 and u_i^2 denote the same entity in the world. Finally, let e_i denote the words in u^1 that also occur in the utterance u_i^2 and let the list $E = \langle e_1, ..., e_r \rangle$ be the evidence for $id(u^1, u_i^2)$. Then preferably Agent 2 should determine the conditional probability: $P(id(u^1, u_i^2) \mid E)$.

The sample space underlying the probability measure consists of instances of mapping problems. Hence, all instances of mapping problems in which the evidence E is present should be considered. Since for determining the joint attention Agent 2 only looks at the corresponding words, $P(e_i \mid id(u^1, u_i^2)) = 1$. So the fact that '3' in one ontology corresponds with the number of children of a person and in the other ontology with the number of courses the same person teaches, plays no role. Moreover, 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 $j \neq k$. Finally, it is clear that $P(e_j \mid id(u^1, u_i^2)) \leq P(e_j)$ if $i \neq j$. Hence,

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

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 and 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 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,

$$
\begin{aligned}
P(e_i) &\approx P(\textstyle\bigwedge_{w \in e_i} \exists j : w \in v(lv_j)) \\
&\approx \textstyle\prod_{w \in e_i} P(\exists j : w \in v(lv_j)).
\end{aligned}
$$

The probability $P(\exists j : w \in v(lv_j))$ that a word w of the utterance u^1 occurs in value of some label-value pair lv_j of u_i^2 is 1 minus the probability that w occurs in no value of any label-value pair lv_j.

$$P(\exists j : w \in lv_j) = 1 - \textstyle\prod_{j=1}^{k} (1 - P(w \in lv_j)).$$

Agent 2 can approximate $P(w \in lv_j)$ 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.

Estimating a value for the probability $P(id(u^1, u_i^2))$ is more difficult since it requires knowledge about the total number of instances of all concepts. Agent 2 may try to estimate this number by analyzing the dependencies between the labels of a concept. Such an analysis would require that Agent 2 has knowledge about many instances of every concept in Ontology 2. Moreover, if the knowledge is available, the analysis would be very time consuming.

An alternative approach is determining the odds that two utterances represent the same entity in the world.

$$
\begin{aligned}
&O(id(u^1, u_i^2) \mid E) \\
&\quad = \frac{P(id(u^1,u_i^2) \mid E)}{P(\neg id(u^1,u_i^2) \mid E)} \\
&\quad \leq \frac{P(id(u^1,u_i^2) \mid E)}{\sum_{j \neq i} P(id(u^1,u_j^2) \mid E)} \\
&\quad \leq \frac{\sum_{j \neq i} P(e_j)}{P(e_i)}
\end{aligned}
$$

The above expression gives an upper bound for the odds that two utterances represent the same entity in the world rather than a lower bound. Nevertheless, this upper bound gives a good indication of the odds if $U^2 \backslash u_i^2$ is a representative sample of all instances of Ontology 2.

Agent 2 has to decide whether two utterances represent the same entity in the world. It uses a threshold value θ^u to make this decision. $O(id(u^1, u_i^2) \mid E)$ must be above θ^u before Agent 2 adds (u^1, u_i^2) to the joint attention. The value of θ^u must be high enough to exclude most coincidental correspondences. If, however, θ^u is too high, there may not be enough evidence even if two utterances represent the same entity in the world. In the experiments presented in Section 5, the threshold value θ^u was set to 1,000,000.

There is one last issue Agent 2 has to take into account when establishing the joint attention JA. The above outlined approach does not work well if concept c_j^2 is a sub-concept of a concept c_k^2. Consider an utterance u_i^2 representing an instance of the concept c_j^2. Since c_j^2 is a sub-concept of c_k^2, there may exist an utterance u_h^2 representing an instance of c_k^2 which has exactly the same evidence e_i as the utterance u_i^2. Clearly, since c_j^2 is a sub-concept of c_k^2, Agent 2 must ignore the evidence for u_h^2. In general, if two utterances u_i^2 and u_h^2 are supported by the same evidence and if the label-value pairs of u_i^2 are a subset of the label-value pairs of u_h^2, then Agent 2 should ignore the evidence for the utterance u_h^2 while determining the odds $O(id(u^1, u_i^2) \mid E)$.

4.4. 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 concepts c^1 is identical to the concept c^2_i, $id(c^1, c^2_i)$, according to one pair in the joint attention and to the concept c^2_j according to another pair. Clearly, assuming a one to one mapping, one of the two must be incorrect.

Let JA^+_i be the pairs in the joint attention JA supporting $id(c^1, c^2_i)$ and let JA^-_i be the pairs in JA supporting some $id(c^1, c^2_j)$ with $j \neq i$. Hence, if $id(c^1, c^2_i)$ holds, the pairs in JA^-_i are incorrectly added to the joint attention by Agent 2. This implies that the probability of $id(u^1_j, u^2_j)$ with $(u^1_j, u^2_j) \in JA^+_i$ given the evidence E_i is *not* conditionally independent of the evidence $\mathcal{E}^-_i = \{E_k \mid (u^1_k, u^2_k) \in JA^-_i\}$. Hence, Agent 2 must take into account that the probability of $id(u^1_j, u^2_j)$ with $(u^1_j, u^2_j) \in JA^+_i$ decreases because of the evidence $\mathcal{E}^-_i$.

Assuming a one to one mapping between corresponding concepts, $P(id(u^1, u^2_j) \mid id(c^1, c^2_i)) = 0$ holds for every $(u^1, u^2_j) \in JA^-_i$. Therefore,

$$\begin{aligned} &P(id(c^1, c^2_i) \mid \mathcal{E}) \\ &\quad \geq P(\textstyle\bigvee_{id(u^1_j,u^2_j)\in JA^+_i} id(u^1_j, u^2_j) \mid \mathcal{E}) \\ &\quad \geq \textstyle\sum_{id(u^1_j,u^2_j)\in JA^+_i} P(id(u^1_j, u^2_j) \mid \mathcal{E}) - \alpha^+_i \end{aligned}$$

Here, $\mathcal{E}$ denotes the combined evidence for all $id(c^1, c^2_k)$. Moreover,

$$\begin{aligned} &P(\neg id(c^1, c^2_i) \mid \mathcal{E}) \\ &\quad \geq P(\textstyle\bigvee_{id(u^1_j,u^2_j)\in JA^-_i} id(u^1_j, u^2_j) \mid \mathcal{E}) \\ &\quad \geq \textstyle\sum_{id(u^1_j,u^2_j)\in JA^-_i} P(id(u^1_j, u^2_j) \mid \mathcal{E}) - \alpha^-_i \end{aligned}$$

Unfortunately, Agent 2 does not know the probability $P(id(u^1_j, u^2_j) \mid \mathcal{E})$. The probability $P(id(u^1_j, u^2_j) \mid \mathcal{E})$ can, however, be eliminated by determining the odds of $id(u^1_j, u^2_j)$ given the evidence. Assuming that $P(id(u^1_j, u^2_j) \mid \mathcal{E})$ is more or less the same for every $id(u^1_j, u^2_j) \in JA^+_i \cup JA^-_i$, and ignoring α^+_i and α^-_i, the odds that $id(c^1, c^2_i)$ holds can be approximated.

$$O(id(c^1, c^2_i) \mid \mathcal{E}) \approx \frac{|JA^+_i|}{|JA^-_i|}$$

Clearly, this is not a good approximations of the odds that two concepts are the same. Nevertheless, it yields good results.

Agent 2 decides whether two concepts are the identical, $id(c^1, c^2_i)$, using a threshold θ^c. To ensure that for each concept c^1 there is at most one concept c^2_i Agent 2 will consider to be identical to c^1, $\theta^c > 1$ must hold.

After having determined the corresponding concepts $id(c^1, c^2_i)$, Agent 2 can remove those pairs (u^1_i, u^2_j) from the joint attention for which $c^2_i \neq c(u^2_j)$. This gives us the following *pruned* joint attention:

$$\begin{aligned} &\widehat{JA} = \\ &\quad \{(u^1_j, u^2_j) \in JA \mid id(c^1_i, c^2_i), c^1_i = c(u^1_j) \Rightarrow c^2_i = c(u^2_j)\} \end{aligned}$$

4.5. Creating a mapping

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 on the basis of the proportion of corresponding words the values of the two labels have in common. Possible associations are:

label(x) ← label(y).
label(x) ← label(y), split(s), first(i).
label(x) ← label(y), split(s), last(j).
label(x) ← label(y), label(z), conc(t).
label(x) ← label(y), split(r), last(j), label(z), conc(t).

Note that the left hand side of the association (←) concerns the destination utterance, and the right hand side one or more source utterances from which words are selected. Also note that the right hand side of an association is in Reverse Polish Notation. The operators used in the associations have the following interpretations:

label(x): the value in the utterance denoted by the label with name: x.
split(s): split the value in to a list using s as separator. Separators are: ' ', ',', ';', and TC (a type change).
first(i): take the i-th element of a list with $i \geq 0$.
last(i): given a list of n elements, take the $(n - i)$-th element of a list with $i \geq 0$.
conc(t): concatenate all the elements on the list and insert t as separator.

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

label(pan) ← label(person.has.street), split(TC), last(0).

Agent 2 searches through a space of possible associations guided by the proportion of words that instances of concepts have in common. Each new element in the joint attention enables Agent 2 to update the strength of the associations. When the joint attention is large enough, Agent 2 may accept certain associations as being correct. Agent 2 has established a complete mapping from one ontology to another when it has a unique association for each attribute in the destination ontology. Note that the mapping is asymmetric and that it only enables communication in one direction. For full communication, Agent 2 must also establish a mapping in the other direction.

Validity An important issue is of course deciding when a proposed association $assoc$ is correct. For a 'label(x)' there might be more than one candidate association of the form: 'label(x) $\leftarrow \ldots$' Agent 2 collects all possible associations for *label*(x) in the set $A(x)$. Clearly at most one association in $A(x)$ can be correct. Using the same approach as presented in Subsection 4.4, an association $assoc$ will be selected from $A(x)$. An association $assoc_i \in A(x)$ is chosen by Agent 2 if the odds

$$O(assoc_i \mid \mathcal{E}) \approx \frac{|\widehat{JA}_i^+|}{|\widehat{JA}_i^-|}$$

are greater than θ^a. Here, $\widehat{JA}_i^+$ are the pairs in the joint attention supporting the association $assoc_i$ and $\widehat{JA}_i^-$ are the pairs in the joint attention supporting an association $assoc_j$ with $j \neq i$.

5. Experiments with ontology mappings

We have evaluated our method for learning an ontology mapping through a large number of experiments. In the experiments we have investigated the number of errors that where made in establishing the joint attention and in creating a mapping. These aspects depend, of course, on the threshold values θ^u, θ^c and θ^a. 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.
- Increasing the number of words in the set W from which the values of an attribute are chosen makes the mapping problem easier.
- The occurrence of sub- and super-concepts makes the mapping problem harder, especially if they differ on only a few labels in an utterance.
- Splitting and concatenating label values makes the mapping problem harder.
- Labels in one ontology that do not occur in the other ontology make the mapping problem harder.

$\|W\|$	recall JA	precision JA	correct associations
25	0%	– %	0%
50	0.7%	66 %	36%
100	23%	94%	87%
250	82%	97%	100%
500	88%	99.7%	100%
1000	90%	95%	100%

Table 1. Experimental results

Since there is no room to report on how the success of leaning a mapping is effected by each combination of the above mentioned dimensions, in this paper we limit ourselves to a number of hard cases that contain most relevant aspects. In our experiments, Ontology 1 consisted of one concept c^1 which had to be map 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.

In each of the experiments, Ontology 1 and 2 were randomly generated, making sure that 10 instances of c^1 corresponded with 10 instances of c^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 and θ^a were chosen. Agent 2 added a pair of utterances to the joint attention JA if the upper bound of the odds was higher than $\theta^u = 1{,}000{,}000$. Moreover, Agent 2 accepted an association as being correct if odds supporting this association were above $\theta^a = \frac{7}{3}$. Finally, the experiments were carried out for different sizes of the set of words W.

In each experiment, we determined the recall and the precision for the joint attention, and counted the number of correct association in a mapping. Table 1 shows the average results over 100 experiments for different numbers of words in W. Notably, no mapping is found if the number of words in W is set to 25. Since two utterances denoting the same instance, have only 5 words in common, the odds will always be lower than $\frac{25^5}{25} = 390{,}625$ if there is at least one other utterance matching only one word. Hence, a lower threshold value θ^u would have been more appropriate for $|W| = 25$, $|W| = 50$, and possibly $|W| = 100$.

6. Conclusions

In this paper, we have presented method for handling the interoperability problem. The method is based on learning a mapping between two ontologies by identifying pairs of instances, one from each ontology, that represent the same entity in the world. An important benefit of this approach is that no domain knowledge is required and that the structure of the two ontologies plays no role.

To guarantee the correctness of the method, estimations for the odds that a mapping is correct have been derived. Moreover the effectiveness of the method has been established through a large number of experiments.

Despite the success, there remain a number of problems. The mappings between different representations of Dutch names such as 'van der Belt' and 'Belt, van der' is not possible without domain knowledge of Dutch family names. Moreover, context dependent mappings such as whether the house number must be placed in front or after the street name depending on the country, cannot be handled. One can show that handling context dependent mappings is an NP-Hard problem.

Future work will aim at extending the method to learn a mapping between groups of inter-related concepts.

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. 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.

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

[6] 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.

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

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

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

[10] 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.

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

[12] 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.

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

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

[15] 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.

[16] 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.

[17] 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.

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

[19] 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.

[20] 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.

[21] 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, 7 1999.

[22] F. Wiesman, N. Roos, and P. Vogt. Automatic ontology mapping for agent communication. In *AAMAS2002*, pages 563–564, 2002.