Matching Unstructured Vocabularies using a Background Ontology

Zharko Aleksovski[1,2], Michel Klein[2], Warner ten Kate[1], and Frank van Harmelen[2]

[1] Philips Research, Eindhoven,
[2] Vrije Universiteit, Amsterdam

Abstract. Existing ontology matching algorithms use a combination of lexical and structural correspondance between source and target ontologies. We present a realistic case-study where both types of overlap are low: matching two unstructured lists of vocabulary used to describe patients at Intensive Care Units in two different hospitals. We show that indeed existing matchers fail on our data. We then discuss the use of background knowledge in ontology matching problems. In particular, we discuss the case where the source and the target ontology are of poor semantics, such as flat lists, and where the background knowledge is of rich semantics, providing extensive descriptions of the properties of the concepts involved. We evaluate our results against a Gold Standard set of matches that we obtained from human experts.

1 Introduction

Semantic integration of heterogeneous datasources is widely regarded as technologically one of the most urgent and scientifically one of the most challenging problems [1–8]. Consequently, much recent work has appeared in this area. In the fields of AI, Knowledge Engineering and Semantic Web research, this problem goes by the name of *ontology matching* (see [1, 6, 7] for a number of recent surveys of this very active field).

According to [6], the methods to solve the problem of ontology matching can be divided into: *terminological* methods which try to identify lexical correspondences between the labels of concepts in the source and target ontologies, *instance-based* methods which use instance data in both source and target ontologies to discover matches [9], *structural* methods which use the structure of the ontologies, and *semantic* methods which use additional logical methods to induce the matches [5].

The majority of the approaches use a combination of terminological and structural methods, where the lexical overlap is used to produce an initial mapping, which is subsequently improved by using the structure of source and target. Hence, this majority of approaches crucially relies on two assumptions:

- sufficient lexical overlap exists between the source and target ontology
- source and target ontology have sufficient structure

In this paper, we will present a case-study where neither of these assumptions hold. In this case study, not only there is insufficient lexical overlap between source and target, but more crucially, the structures to be matched contain *no structure at all*: they are simply lists of terms, instead of richly structured ontologies. Consequently, current state-of-the-art matchers are expected to fail.

We believe that our case-study is representative of many realistic cases. Experience with Semantic Web applications shows that many of them rely on rather lightweight semantic structures, providing at most a hierarchy of terms, where often this hierarchy is only a 2-3 levels deep [10]. Hence, the reliance of existing ontology matchers on such structure is indeed an important limiting factor.

After showing indeed the failure of a number of state-of-the-art matchers in our case-study, we present a novel method for ontology matching that uses an additional source of *background knowledge* to compensate for the lack of structure to be found in source and target vocabularies as well as the freedom in choice of terminology.

The basic idea of our approach is to first align the concepts from the source and target ontologies with the background knowledge, then use the structure of this background knowledge to derive semantic relationships between the source and target concepts, and finally use these relationships to induce a mapping between them.

We built a system that implements this approach and tested it with the data from our case-study. We then score the performance of our system against a human-created Gold Standard, and show that on such poorly structured data, our system performs significantly better than existing state-of-the-art matchers.

In the following sections we first describe the details of our case (section 2), and we show the low success rate of some existing state-of-the-art matchers on these data (section 3). We then proceed to explain our own matching process, based on the use of background knowledge (section 4 and 5), and show in section 6) that we achieve considerably higher success rates than the existing matchers. The final section 7) concludes.

2 The data in our case study

In this section we describe the data involved in our case study. The challenge was to match two vocabularies that were taken from the medical domain. The vocabularies are lists of reasons for admission in the Intensive Care Unit (ICU) of two Amsterdam hospitals. A reason for admission describes a problem, why a patient was brought into the ICU. Each patient arriving at the ICU in either hospital is classified using one or multiple terms from the corresponding list. These classifications are used for monitoring patient progress, for planning of required ICU resources, and for off-line nationwide quality comparison of different ICU's.

Source vocabulary: As source vocabulary for our mapping case-study, we use a set of reasons for admission from the OLVG hospital in Amsterdam. It is a flat list of terms with no structure. The list is partly based on the ICD-9-cm[3] vocabulary, and on the Dutch "Classificatie van Medisch Specialistische Verrichtingen" (CMSV)[4], a classification of medical procedures. During its use in the past three years, the OLVG list has been extended with additional descriptions of medical conditions of patients

[3] `http://www.cdc.gov/nchs/about/otheract/icd9/abticd9.htm`
[4] `http://www.nictiz.nl/kr_nictiz/2527`

at the ICU. The resulting list is a mixture of problem descriptions at several levels of abstraction with minor redundancy. It does not only contain reasons for admission to the ICU, but also other medical conditions that are relevant during the stay of patients at the ICU. We limited our experiments to the list-elements actually used at admission, identified as those terms that are used for describing patients during the first 24 hour of their stay. The resulting list contains 1399 problem descriptions consisting of maximal 7 words each. 95% of these descriptions consist of no more than three words. The list is mainly in Dutch but also contains English terms.

Target vocabulary: As target vocabulary, we use a second set of reasons for admission, used at the AMC hospital in Amsterdam. The AMC vocabulary consists of a flat list of 1460 reasons for admission at the intensive care unit of the AMC hospital. This list was used as the target vocabulary in our experiments.

Background ontology: As outlined above, the essential idea of our aproach is to first align the unstructured source and target vocabularies with a given background ontology, and then to use the structure of the background ontology to derive matching relationships between the source and target vocabularies. In our case study, we use the DICE ontology as background knowledge. DICE has been developed by the Medical Informatics group at the AMC hospital. It is a medical terminology, formalized in OWL DL[5], of some 2300 concepts, described by some 5000 lexical terms. These concepts are related to each other with some 4300 relational links of 50 different relation types. DICE mainly aims to cover concepts in the Intensive Care domain, and is structured in five different hierarchies (called "aspects" in DICE): abnormalities (255 concepts), medical procedures (55 concepts), anatomical locations (1512 concepts), body subsystem (13 concepts), and causes (85 concepts). Together, these five vocabularies are the main organisational structure of DICE. Each aspect has a domain of possible values, organized in a tree structured taxonomy. The concepts in the aspect taxonomies are labeled with a language attribute, in the current version either Dutch or English. If a concept is named with multiple terms, one of the terms functions as 'preferred' term - label, and the others as synonyms.

3 Performance of state-of-the-art tools on our case study

3.1 Ontology alignment tools

There are several other approaches for ontology matching. An overview can be found in [6, 1, 7]. In this section, we summarize three of the most prominent approaches.

- FOAM is an ontology alignment framework to fully or semi-automatically align two or more OWL ontologies, developed by the university of Karlsruhe [11]. It is based on heuristics (similarity) of the individual entities (concepts, relations, and instances). As result, it returns pairs of aligned entities. It can handle ontologies within the DLP-fragment of OWL. Part of FOAM is a machine learning component that optionally takes user feedback into account.

[5] `http://www.w3.org/TR/owl-guide/`

– Falcon-AO [12] is an automatic ontology matching tool, developed by the South East University of China. It outperformed all other ontology matchers in the 2005 ontology alignment initiative [13]. Falcon-AO regards ontologies as graph-like structures, and then produces mappings between elements in the two graphs that correspond semantically to each other. Both of linguistic similarity and structural similarity are taken into account. There are two matchers integrated in Falcon-AO: LMO for syntactic comparison based on edit distance, and GMO for graph-based comparison.
– S-Match is a algorithm and tool developed by the University of Trento [14], based on CTXmatch[5]. S-Match takes two trees, and for any pair of nodes from the two trees, it computes the strongest semantic relation holding between the concepts of the two nodes. For this, it uses lexical techniques, background knowledge in the form of relations between synsets in WordNet, and the structure of the tree. S-Match is restricted to tree-like structures used for classification purposes.

As is already illustrated in the descriptions above, most alignment tools exploits a combination of syntactical comparison techniques and structural comparison techniques. This mixed approach can work well in practice, as the results in the ontology alignment initiative show. However, when one of the sources to be matched merely is a unstructured list, the mixed approaches reduce to lexical comparison of labels.

Only one of the tools, i.e. S-Match, exploits background knowledge to do the mapping. However, in the current version S-Match can only use a predefined set of background knowledge sources, such as Wordnet and UMLS. Moreover, it only uses the class hierarchy of background ontologies.

3.2 Performance of other tools

We have applied two of the ontology matching tools described above to the data that is described in our case study. We loaded an OWL representation of both the OLVG list and the AMC list plus the DICE background knowledge into the tools. In this section we only describe the amount of matches that we found, in section 6.1 we give an analysis of the correctness of these matches.

When using FOAM to align the two lists, we initially got 326 matches, but those included symmetric matches. Effectively FOAM found 159 matches. An analysis revealed that many obvious matches were missing because synonym labels were ignored. To solve this, we have regenerated the AMC list with separate concept definitions for each known synonym. This resulted in 696 effective matches.

We also used Falcon-AO.[6]. Initially, the files were too large to run in one step. After splitting the AMC-ontology in several files, we were able to run the tool, and extrapolated that we would end up with less than 100 matches. However, because all matches were necessary based based on lexical measures only, we tried to use the lexical matcher component (LMO) in a stand-alone configuration. This has two advantages. First, the stand-alone LMO matcher is much more efficient, so that matching can run on the complete AMC ontology. Second, it returns a a ranking of all matches found, and not

[6] Experiments performed in collaboration with Dr. Wei Hu, South East University of China.

just the ones that are above the threshold. As a result, we get much more matches, but many of them with a low confidence level. When using the stand-alone LMO matcher, 683 matches were returned.

Unfortunately, because S-Match isn't freely available, we haven't yet been able to use S-Match on our dataset with a medical thesaurus as background knowledge.

4 Semantic matching of concepts from unstructured vocabularies

The scheme of our matching approach is depicted in Figure 1. The task is to match the source vocabulary to the target vocabulary, where both vocabularies are unstructured (flat) lists. Our approach consists of finding semantic matches by using a (richly structured) ontology that holds background knowledge about the domain. First, the source and target vocabulary are each matched with the background knowledge ontology producing so-called *anchoring matches*. Anchoring matches connect a source or target concept to one or more concepts in the background knowledge ontology, which we call anchors. Then, based on the relationships among the anchors entailed by the background knowledge, we induce in a second step how the concepts from the source are matched to the concepts in the target vocabulary, yielding the *semantic match* we are looking for. A concept can match to several anchors, in which case the collection of anchors is combined according to the semantics of the background knowledge.

In the following, we will describe the anchoring process (subsection 4.1) and the process of inducing the source-to-target semantic match (subsection 4.2) in more detail.

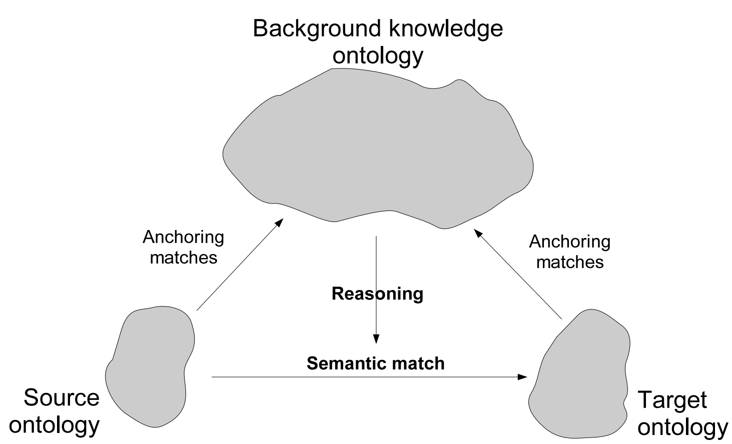

Fig. 1. Our general approach: Matching a source vocabulary to a target vocabulary using an ontology describing background knowledge.

4.1 Anchoring vocabulary concepts with the background knowledge

As explained, the source and target vocabularies are anchored to concepts in the background ontology. The anchoring matches can be established manually by a domain expert, or automatically using existing concept matching techniques.

The automatic anchoring in our approach, is performed by a simple lexical heuristic that discovers partial matches between two strings. It makes use of the concept's labels and synonyms only. A concept's label is the string name assigned to the concept. Quite often, these labels come with a list of synonyms for the concept. We used both the labels and the synonyms in our comparison. Our heuristic is based on comparing the number of matching words: If all the words in a label or synonym of a concept A are found in a label or synonym of concept B, it concludes that A is an anchor of B. We also used some simple Dutch morphological rules to deal with the common Germanic construction of compound words that do not have a delimiting space between the words. For example, *"hersentumor"* (brain tumor) is a special case of *"tumor"*.

Figure 2 depicts an example of discovering an anchoring match.

Clearly, these heuristics are very simple, and it is not difficult to think of cases where they fail: In Figure 2, for example, *"long brain tumour"* would also end up being a special case of *"brain"*.

It would seem that such a simplistic lexical mapping would not suffice, and it would seem that we are replacing one ontology matching problem (source to target) by two ontology matching problems (anchoring source and target to the background knowledge). Why then, would this make our problem any easier? The surprising thing is, as revealed in our experiments, that this is indeed the case. After presenting our experimental data in the next section, we will discuss why the use of such simple lexical heuristics is sufficient for the anchoring process, while it is *not* sufficient for the direct source-to-target matching.

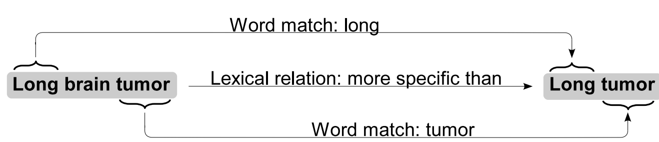

Fig. 2. An example of lexical anchoring by comparing two labels: *"Long brain tumor"* and *"Long tumor"*. The first consists of a superset of the words from the second label, so the second can be considered a property value filler of the first.

The anchoring process implicitly adds structure to the unstructured source and target vocabularies, and has established relations between source and target concepts through the relations between their anchors in the background ontology. We exploit this structure in the next step, to discover semantic matches between the source and target vocabulary concepts.

4.2 Semantic matching - the use of background knowledge

As said, we make use of the background knowledge ontology to perform the discoveries of semantic matches. When comparing two concepts A and B having anchors A' and B' respectively, we compare A' and B', and, if they are related, infer that A and B are related as well.

An example is given in Figure 3: the concept *"Dissection of artery"* is found to have location *"Artery"*, and the concept *"Aorta thoracalis dissection"* is found to have location *"Aorta thoracalis"*. A relation is inferred between these two medical concepts, since they describe related anatomical locations: according to the background ontology *"Aorta thoracalis"* is a kind of *"Artery"*. Hence, the source concept can be inferred to have a more specific location than the target concept. Notice that with pure lexical methods, no meaningful match between these two concepts could have been derived. The use of background knowledge was essential to derive this match.

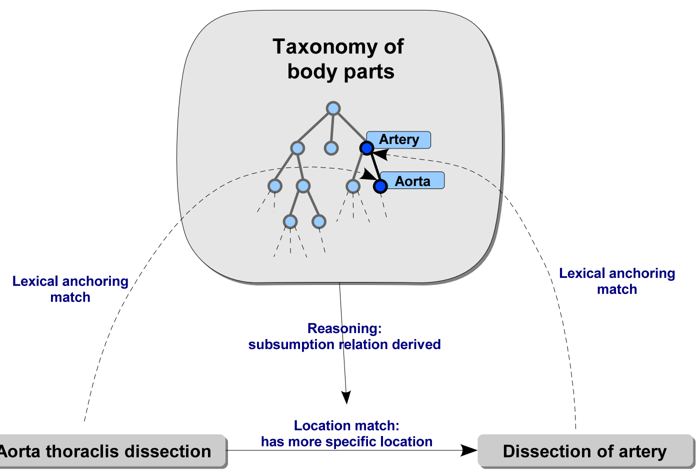

Fig. 3. An example of relating two medical concepts using background knowledge. A semantic match is discovered using the location taxonomy.

In general, either the source concept A or the target concept B could be anchored to multiple anchors and the background knowledge could reveal relationships on anchors that represent different properties. On the one hand, this makes the comparison more complex. On the other hand, if multiple anchors are related in similar ways, they reinforce that the main concepts A and B are related in the same way. In case the multiple anchors are related in incompatible ways (e.g. anchor A'_1 subsumes anchor B'_1, but anchor A'_2 is subsumed by anchor B'_2), a subsumption relation between the concepts cannot be inferred. However, they do reveal the source and target concepts have some relationship and are within some semantic distance.

This is illustrated in the example depicted in Figure 4, where we try to match the concepts *"Heroin intoxicatie"* and *"Drugs overdosis"*. According to the background knowledge, *"Heroin"* is a kind of *"Drugs"*, while *"Overdosis"* is a kind of *"Intoxicatie"*, i.e. the two aspects have a subsumption relationship to each other, however, in reverse direction between the concepts. Hence, the concepts are neither equivalent nor one subsuming the other. However, the two concepts do have a big semantic overlap. Again, note that these concepts do not have any lexical similarity - their describing labels consist of entirely disjoint sets of words. It was only possible to discover this match by using the background knowledge.

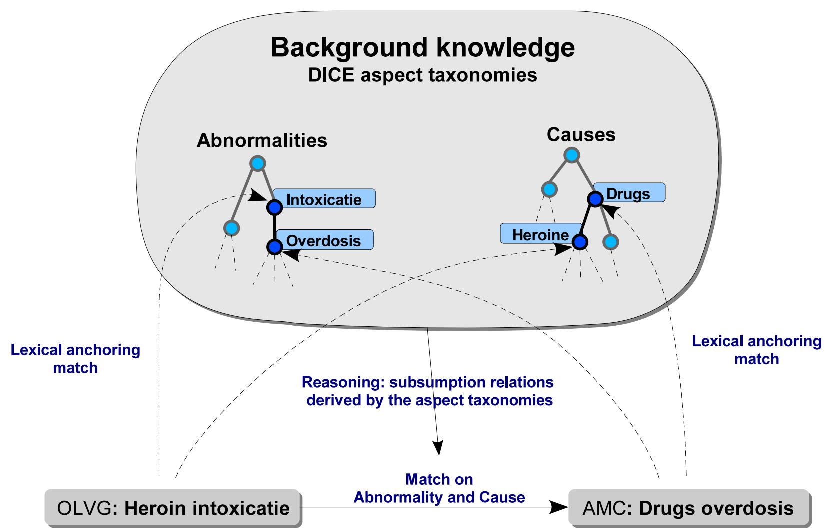

Fig. 4. An example of matching two concepts using background knowledge. The concepts are not equivalent but do have a big semantic overlap.

The successful application of the method depends on the richness of the background knowledge. As discussed later in Section 6, with increasing richness, the more likely it becomes that anchoring matches can be established in the first place, but, more importantly, the more likely it becomes a relation between the anchors can be found.

Experience in practice has shown that concepts from two matching ontologies are rarely precisely equivalent, but rather have some (otherwise unspecified) semantic overlap. Consequently, finding such semantic relationships seems more useful for integration purposes, than finding precise equivalences.

4.3 Comparison with other approaches

Our approach can be compared with the semantic coordination approach proposed by Bouquet et al. [5]. That approach assumes the source and target vocabularies do have some structure and each concept does have a label that is meaningful in natural language. It proceeds in two phases. In the first phase, called explicitation, the concepts from source and target ontology are transformed into propositional expressions, using the labels and surrounding structure such as ancestor and sibling labels. The words in the label are considered as propositional atoms. An additional source, such as WordNet[7], may be used to enrich the explicitation, by taking the senses returned by WordNet on each word in the label as the propositional atoms. In the second phase, the obtained propositional expressions are tested whether one implies the other, for example, in a SAT solver. Since the propositional expressions capture the semantics of the original concepts, valid implications indicate a semantic subsumption relation between the concepts.

In our scheme, we also assume the labels are meaningful in natural language. However, we do not assume a (semantic) structure to be present. We also create logic expressions and subsequently evaluate them, however, in our case, the logic framework is given by the background knowledge.

Usually, the description in a background knowledge ontology are expressed in a logic richer than propositional logic. Accordingly, after the anchoring (which may be compared with the "explicitation" phase) we obtain richer logic representations. In particular, we include concept relations. In line with frame-based systems, quite often concept relations have the character of properties: one concept is the filler of a property of the other concept. In our reasoning paradigm we took the fillers apart in a separate classification and combined the different classifications of all fillers to derive the match between the main concepts.

5 Experiments to test our approach

In our experiments we matched the source and target lists (i.e. the unstructured OLVG and AMC vocabularies) both lexically, i.e. directly, and semantically, i.e. using the DICE ontology as background knowledge.

Experiment 1: Lexical matching. In the lexical match, we directly matched pairs of terms from the two vocabularies, using the lexical matching method described in Section 4.1. In testing for equality of terms, we allowed for edit-distance of two characters using Levenshtein string distance [15], to compensate for the typing mistakes in the lists. The result is a list of pairs of terms, that were either equivalent or related in a more-general-than relation.
Hypothesis: We expect that the results of this lexical matching step are comparable to the performance of existing tools on this data, such as discussed in Section 3, since on these data the existing tools are also reduced to performing lexical matching only.

Experiment 2: Semantic matching. When matching the vocabularies semantically, we followed the general scheme of our approach, depicted in Figure 1, using the five aspect taxonomies in DICE as the background knowledge. First, in the anchoring step, we lexically matched the terms from the OLVG and AMC source and target vocabularies to

[7] `http://wordnet.princeton.edu/`

each of the five DICE aspects taxonomies, producing anchors in the background ontology for the terms from the vocabularies. For the AMC-DICE anchoring, besides anchoring through the lexical matching procedure, we also used a given anchoring schema that was created manually by experts from the AMC hospital. After obtaining the anchors we used the relationships specfied between the anchors in the DICE taxonomies to infer a semantic match between the source and target concepts. Figure 5 depicts the scheme (which is essentially the general scheme from Figure 1 instantiated for our experiment). Note that each term from source and target ontology is matched multiple times to the background ontology, viz. once per aspect taxonomy.
Hypothesis: We expect the results of semantic matching to be better than the results of direct lexical matching between the two vocabularies.

Evaluation: For measuring the performance of the various methods and tools, we created a Gold Standard solution for this problem. A medical expert was invited to create manually matches between the OLVG and AMC vocabularies. He was given a random sample set of 200 concepts from the OLVG list, for which he was asked to find the matching concepts in the AMC list. This set was created as follows: 30 terms were selected for which we knew that good anchorings in DICE would be available, together with the top 15 most frequently used terms from the OLVG vocabulary, supplemented with randomly drawn terms to a total of 200 terms. For these 200 terms, the expert created matches for 125 concepts, leaving the other 75 “unknown”. For each of the matched OLVG concepts he proposed one AMC concept as the most appropriate match. No statements about alternates were made. This yields a Gold Standard on the order of about 10% of the entire vocabulary, which is sufficient for reliable performance measurements.

6 Results

6.1 Experiment 1: Lexical matching.

We will first present the results of our “baseline experiment”: directly trying to find lexical matches between the OLVG and AMC vocabularies without the mediating role of the DICE background knowledge. When matching the OLVG list to the AMC list directly, using the lexical technique only, 582 OLVG concepts were matched to concepts in the AMC list. Of these, 274 were found to be lexically equivalent, and the remaining 308 concepts were partial matches.

Evaluation against the Gold Standard The 582 OLVG terms for which lexical matches were produced represent some 42% of all OLVG terms. Comparison against the Gold Standard revealed that some 24% were correct. In section 3 we already reported on the number of matches found by tools like FOAM and Falcon-AO. We also scored the precision of these tools against the Gold Standard. These figures are summarised in the table in Figure 6.

[8] These figures are the numbers of correct matches on those matches for which the Gold Standard contained an answer.

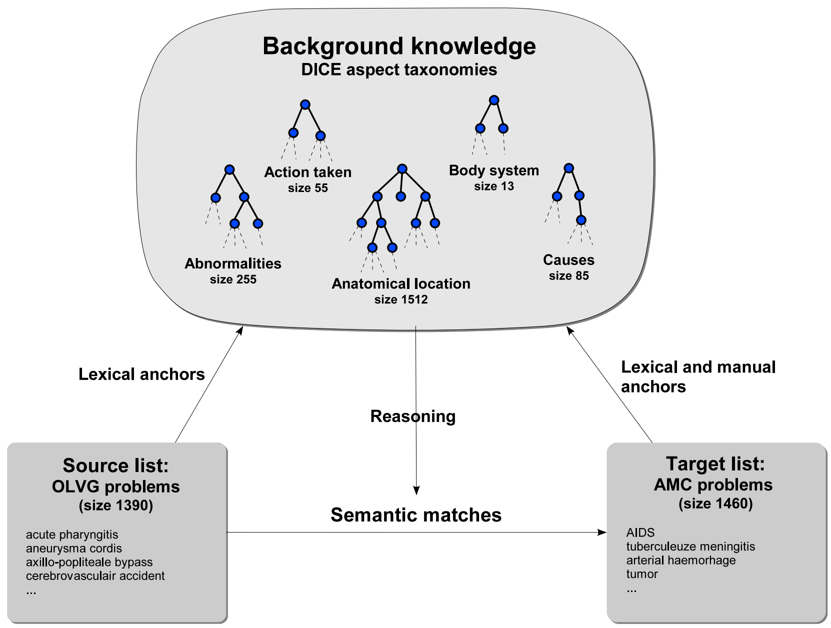

Fig. 5. Matching the OLVG and AMC vocabularies using the aspect taxonomies of DICE as background knowledge ontology (this is Figure 1 instantiated for our experiments.)

A manual analysis revealed that around 260 concepts in the OLVG list (i.e. 19% of the corpus) have a large lexical overlap with concepts in the AMC list. The figures in the table show that both FOAM and our lexical matching method finds a comparable percentage of correct matches. A possible explanation for the fact that our method scores a higher percentage of correctness (24%) than the estimated lexical overlap (19%) is that the Gold Standard is slightly biased towards frequently occurring problems (see the description of the creation of the Gold Standard in the previous section), for which the lexical overlap can be higher.

The LMO module o Falcon-AO retrieves much fewer of correct matches. The most likely explanation of this effect is that Falcon-AO limits itself to 1-1 matches, preventing many plausible partial matches. This is illustrated by the following example matches

	total matches found on corpus (n=1399)	correct matches found[8] on Gold Standard (n=200)
lexical matching	582	49 (=24%)
FOAM	696	41 (=20%)
Falcon-AO	683	28 (=14%)

Fig. 6. Results of the lexical matching experiment

produced by Falcon-AO on our case-study data:

```
OLVG#Oesofagus_perforatie ↦ AMC#Oesofagus ruptuur
OLVG#Oesophagus_resectie  ↦ AMC#Oesofagus perforatie
```

although both by themselves reasonable, together these matches prevent the obvious match

```
OLVG#Oesofagus_perforatie ↦ AMC#Oesofagus perforatie
```

because Falcon limits itself to 1-1 matches: both of these terms are already part of another match. The obvious solution would be to allow terms to participate in multiple matches, producing an n-m matching as done by our lexical method.

6.2 Experiment 2: Semantic matching

In the first step of semantic matching OLVG and AMC concepts are anchored into the DICE background ontology.

Experiment 2: Semantic matching, anchoring step. When anchoring the OLVG vocabularies to DICE, we used the lexical technique described earlier, and we found in total 549 of OLVG concepts anchored to DICE concepts, via 1298 anchors.

For anchoring the AMC vocabulary to DICE we used a combined expert- and automatic approach. An expert manually established 4568 DICE-anchors for the AMC vocabulary. We enhanced these anchor matches using the lexical matching technique and found a further 1248 new anchors, which increased the amount of anchors to a total of 5816, anchoring a total of 1404 concepts.

Notice that the anchorings are many-to-many relations: a single term from source or target vocabulary can have multiple anchor terms in DICE, either in a single or in different aspect taxonomies. Table 7 shows how many terms were anchored, and how often our lexical heuristics were able to establish anchorings to multiple DICE aspect taxonomies. Such anchoring in multiple aspects is important, because it will enable the inference step to use multiple DICE taxonomies to infer potential semantic matchings.

Table 7 shows that our simple lexical heuristics succeeded in constructing anchors for 39% of the OLVG vocabulary. This indicates that indeed our weak lexical heuristics are able to establish anchorings, to be used in the second step of our approach. The high percentage of anchoring for the AMC vocabulary is due to the contribution of the manually constructed anchors.

Figure 8 details how the anchors are distributed over the five DICE aspects (separate taxonomies). It shows that the anchors are very unevenly distributed over the various aspects (with only three anchors established from the OLVG vocabulary to aspect hierarchy on body-systems), and a similarly uneven relative contribution between expert-created and lexically found anchors across the different aspects (with again the body-systems aspect producing very few lexical anchors for the AMC vocabulary).

[9] = 4568 manually + 1248 by lexical matching

	OLVG	AMC
anchored on 5 DICE aspects	0	2
anchored on 4 DICE aspects	0	198
anchored on 3 DICE aspects	4	711
anchored on 2 DICE aspects	144	285
anchored on 1 DICE aspect	401	208
total nr. of anchored terms	549 (=39%)	1404 (=96%)
total nr. of anchoring relations	1298	5816[9]

Fig. 7. Results of the anchoring step in our experiment

		AMC list			OLVG list
Aspect	Expert-manual	Additional lexical	Total	Lexical
Abnormality	1168	271	1439	354
Action taken	292	122	414	109
Body system	1217	2	1219	3
Location	1336	721	2057	255
Cause	555	132	687	60
	4568	1248	5816	781

Fig. 8. Distribution of anchors over the different DICE aspect taxonomies

Experiment 2: Semantic matching, inference step. In the second step, relationships between anchors in DICE are used to infer matches between source and target terms. As a result of this matching, a matching AMC term was derived for 538 OLVG terms. Of these, 413 matches were based on inference in a single DICE aspect, while 135 matches were supported by inference in two aspects (i.e. the inference in two DICE taxonomies produced support for the same match).

Evaluation against the Gold Standard The evaluation of semantic matches against the Gold Standard is made more complicated by the fact that the n-m anchors can also produce n-m matches between source and target vocabularies. When a single OLVG concept matches with multiple concepts in the AMC vocabulary, we ranked the matchings as follows:

1. If the match corresponds to a direct, lexical equivalence, it is ranked highest in the result set.
2. The remaining matches, were ranked according to the number of DICE aspect taxonomies that supported the match
3. Matches based on the same number of DICE aspects, were ranked according to the number of equivalence matches on DICE properties (ie preferring equivalences over part-of, contained-in, type-of, etc).

We assess the performance of our method on the Gold Standard containing human created matches for a random set of 200 OLVG concepts. For 65 concepts, our method produced the same results as the expert, proposing the matched AMC concept as a single best candidate. For 8 concepts, our method found the expert match in the first five suggested matches. For 43 concepts, neither our method, nor the expert produced any matching. For 49 concepts matched by the expert our method did not produce any, or produced matches of low confidence. For the remaining 35 concepts, the matches produced by our method were different from those by the expert. Manual inspection revealed that our method produced either new matches for which the expert did not produce any, but that did seem plausible, or other matches that seem a refinement of the proposed expert matches.

These figures are summarised in the table in Figure 9, including the results for the lexical approach reported already in Figure 6 above. This summary shows that on such semantically empoverished vocabularies as in our case-study, the use of semantic background knowledge as part of the matching can substantially improve both the number of matches found and the quality of these matches, as compared to either a purely lexical technique, or as compared to existing tools that, in the absence of any structure in the source and target vocabulary, default to lexical matching only.

	Semantic matching	Own Lexical matching	FOAM	Falcon-AO
agreement on single best match	65 (=32%)	43	35	22
agreement among top 5 matches	8 (= 4%)			
agreement on no match possible	43 (=22%)	43	26	32
improvement over expert match	35 (18%)	6	6	6
TOTAL POSITIVE:	**151(=76%)**	**92 (=46%)**	**67 (=33%)**	**60 (=30%)**
wrong match found		5	47	78
incorrectly found no match	49(=24%)	103	86	62
TOTAL NEGATIVE:	**49(=24%)**	**108 (=54%)**	**133 (=67%)**	**140 (=70%)**

Fig. 9. Summary of evaluation on Gold Standard (n=200)

7 Conclusion

We explored the use of a semantically rich background knowledge in semantic matching of semantically poor concept lists. We provided empirical evidence that background knowledge can improve the matching process considerably. The use of a background knowledge source is the only way to discover matches, when there is no terminological, instance or structural match between the matching ontologies.

Work by [5] has already shown the usefulness of simple background knowledge in the form of the WordNet hierarchy. We extended the method of [5] by using a much richer source of background knowledge. This enabled us to reason across multiple hierarchies in the background knowledge, and made it possible to discover relations between concepts which were not directly related in a subsumption relation.

8 Future work

In future work we will focus upon testing with ontologies of larger size. Such tests can provide for stronger evidence whether this method can be successfully applied to the ontology integration problem. We are currently setting up experiments on mapping the anatomical subhierarchies of CRISP and MeSH, while using FMA as the background knowledge source, and using parts of the UMLS metathesaurus as the Gold Standard.

An interesting further question is how the number and quality of matches found increases with a growth in background knowledge. We are currently performing experiments on the same case-study from this paper (mapping the OLVG and AMC unstructured vocabularies), but while using increasingly more background knowledge. We are currently redoing the experiments from this paper, but by adding such sources as ICD10 and MeSH as background knowledge besides DICE.

Acknowledgements

This research was partly supported by the Netherlands Organisation for Scientific Research (NWO) under project number 634.000.020. We wish to thank to the expert for providing the reference matches and the OLVG and AMC hospitals for providing us with the lists.

References

1. Rahm, E., Bernstein, P.A.: A Survey of Approaches to Automatic Schema Matching. VLDB Journal **10**(4) (2001)
2. Doan, A., Madhavan, J., Domingos, P., Halevy, A.: Learning to Map Between Ontologies on the Semantic Web. In: WWW '02: Proceedings of the Eleventh International Conference on World Wide Web, ACM Press (2002) 662–673
3. Noy, N.F., Musen, M.A.: PROMPT: Algorithm and Tool for Automated Ontology Merging and Alignment. In: Proceedings of the Seventeenth National Conference on Artificial Intelligence (AAAI-2000), Austin, TX, AAAI/MIT Press (2000)
4. McGuinness, D.L., Fikes, R., Rice, J., Wilder, S.: The Chimaera Ontology Environment. In: Proceedings of the Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on Innovative Applications of Artificial Intelligence, AAAI Press / The MIT Press (2000) 1123–1124
5. Bouquet, P., Serafini, L., Zanobini, S.: Semantic coordination: A new approach and an application. Technical report, University of Trento (2003)
6. Euzenat, J., Bach, T.L., Barrasa, J., Bouquet, P., Bo, J.D., Dieng, R., Ehrig, M., Hauswirth, M., Jarrar, M., Lara, R., Maynard, D., Napoli, A., Stamou, G., Stuckenschmidt, H., Shvaiko, P., Tessaris, S., Acker, S.V., Zaihrayeu, I.: Survey of scalability techniques for reasoning with ontologies. KnowledgeWeb Project deliverable D2.1.1 (2004)
7. de Bruijn, J., Martin-Recuerda, F., Manov, D., Ehrig, M.: D4.2.1 state-of-the-art-survey on ontology merging and aligning v1. SEKT Project deliverable D4.2.1 (2004)
8. Guarino, N., Masolo, C., Vetere, G.: Ontoseek: Content-based access to the web. IEEE Intelligent Systems 1094-7167/99 (1999)
9. Ichise, R., Takeda, H., Honiden, S.: Integrating multiple internet directories by instance-based learning. In: Proceedings of the eighteenth International Joint Conference on Artificial Intelligence. (2003)

10. Schlobach, S.: Semantic clarification by pinpointing. In: Proceedings of the second European Semantic Web conference, LNCS. Springer Verlag (2004)
11. Ehrig, M., Sure, Y.: Foam - framework for ontology alignment and mapping; results of the ontology alignment initiative. In Ashpole, B., Ehrig, M., Euzenat, J., Stuckenschmidt, H., eds.: Proceedings of the Workshop on Integrating Ontologies. Volume 156., CEUR-WS.org (2005) 72–76
12. Jian, N., Hu, W., Cheng, G., Qu, Y.: Falcon-ao: Aligning ontologies with falcon. In: K-Cap 2005 Workshop on Integrating Ontologies. (2005)
13. Euzenat, J., Stuckenschmidt, H., Yatskevich, M.: Introduction to the ontology alignment evaluation 2005. Technical report (2005)
14. Giunchiglia, F., Shvaiko, P., Yatskevich, M.: S-match: an algorithm and an implementation of semantic matching. In: Proceedings of the European Semantic Web Symposium (ESWC). (2004) 61–75
15. Gusfield, D.: Algorithms on Strings, Trees, and Sequences: Computer Science and Computational Biology. Cambridge University Press; 1st edition (1997)