Mapping Ontologies using SAT Solvers

Lee Parnell Thompson

April 21, 2011

Abstract

In this project the creation of an algorithm created to discover the relationships between two ontologies, based off the paper "A SAT-Based Algorithm for Context Matching" by Bouquet. The project entails two primary sections, a semantic explicitation phase and a semantic comparison phase. The first phase takes an ontology and adds additional information based on the structure and label contents. This is accomplished through a lexical analysis of the labels using WordNet. The added information provides logical inter-relationships and also provides a common vocabulary between differing ontologies. The logical information between two ontologies are then used to compare and draw logical inferences between them to create a single merged ontology. Once completed the merged ontology is then formed as a set of first order logical rules. Secondly the logical rules are converted into CNF form where a SAT solver is used to determine satisfiability of the merged ontology and any further queries onto it.

1 Introduction

The World Wide Web is a global space filled with information in the form of interlinked documents accessible through the Internet. Using a browser users can view the documents and access the information on them, which is usually in textual form. These documents provide links to other documents in the form of universal resource locators(URLs) which provide the ability to jump from the current document to another. To find a particular document search engines are used which parse the information in the documents and use algorithms to present the best document match when given a user query. The WWW can be seen as a web of unstructured information, the idea of the semantic web is to create a web of structured information.

The idea of the *semantic web* was proposed by Tim Berners-Lee, also credited for creating the internet, who coined the term in 1999. In the book Berners-Lee likened the semantic web to a "Web of Data" that would allow for easier access to information for both humans, and for machines. A problem with the current web is the naive way in which information is organized, if the web data was stored in some sort of structured manner then this would facilitate finding information and related information for both humans and machines. Data would be presented on the semantic web using *RDF(Resource Description Framework)* format. Unlike

HTML a document in RDF contains triples of information, $< subject, predicate, object >$. These triples can be seen as a labeled graph, $< node_1, edge, node_2 >$. So the information "a guitarist is a musician who plays the guitar" and "musicians are artists who create music" could be represented by the labeled graph. Now not only can the information be more easily traversed, but logical queries can be applied onto the graph, for example the query "do guitarists create music?" can be answered by knowing that musicians create music, and guitarists are musicians.
$< musician, subgroup, artist >$
$< musician, create, music >$
$< guitarist, subgroup, musician >$
$< guitarist, plays, guitar >$
Not only is the data inside the semantic web structured but it also contains unique identifiers for data. Like a uniform resource locator which is an unique identifier for documents, data in the semantic web would contain a uniform resource identifier(URI), which would enable that particular data to be accessed anywhere.

Ontologies are the formal representation of concepts and the relationships between those concepts. While this is the computer science interpretation, the word Ontology descends from Greek (*onto*: of that which is; *-logy*: science) and was, until recently, used in a purely philosophical context as the study of being/existence. Part of this field deals with categorizing the different types of being and determining how these types logically relate to each other, which is most likely why the term ontology has been taken up by computer science. The earliest computer science paper that I have found that utilizes the word ontology defines it thusly "An ontology is an explicit specification of a conceptualization... For knowledge-based systems, what "exists" is exactly that which can be represented." [10]. In this paper the word Ontology will mean a formal representation of concepts,and the relationship between those concepts, inside a domain– which can be represented as a graph of nodes(concepts) and edges (relationships), or see the definitions below for more detail.

What is important about ontologies is the same as the semantic web, the ability to traverse the data graph to allow for easy information retrieval. Ontologies are currently used in many areas, and are becoming increasingly popular. For example, one of the largest ontologies is the Gene Ontology [1]. The Gene Ontology gives biologists a controlled vocabulary that they can use to publish information about genes, and gene products, which become part of the large graph of information. Once the information is inside the Gene Ontology others can now retrieve that information through logical queries on the graph, and provide further information to the community. Other major ontologies are WordNet and DBPedia [2, 13]. Wordnet provides a structured ontology for words and their meanings while dbpedia is an attempt to provide all the information of wikipedia in the format of the semantic web.

While ontologies provide a lot of power to organize concepts a large drawback is that ontologies are created by different groups of people. Each group has their own conceptualization of the world and will model their ontology differently, even if the domain is the same. For example given the domain of music, two very different ontologies could be created, depending on the group. This makes it hard, or impossible, to perform queries across multiple ontologies, or to find the relationships between these ontologies, a term called *semantic interoperability.* In order for multiple ontologies to be used either the ontologies must be mapped to a single "authoritative" ontology, or the ontologies must be mapped to each other. Most often this mapping is accomplished by a simple syntactical edit distance based approach. For example let C be the set of concepts in one ontology and C' the concepts in another. A simple edit based approach would create a pairwise distance matrix between all the concepts in C and C' and assume correlation between any concept that was greater than some threshold. While this is actually common in practice there are more intelligent ways of performing this mapping.

In ontologies the meaning of a label depends not only on its spelling but also the context in which it occurs. For example the two words stock car, and car stock, mean very different things but have identical words. This means that to perform a correct mapping between ontologies we must provide additional information. In the paper by Bouquet [6] a syntactical mapping is performed but is extended with other meta-data information to provide a more intelligent map. Meta-data is any additional information that provides more information about data or the structure of data, "Data about Data". The ontology graph, and the set of attributes that provide meta-data about the graph, are called a context. This meta-data is found through the use of wordnet and using its built in knowledge of relationships between words to provide structure. With the addition of this meta-data it is possible to convert these contexts into a form that is possible to solve via a SAT solver.

2 Body

Definition 1. *Ontology*

An ontology is the representation of concepts, and the relationship between those concepts, as a labeled graph. More formally it is a triple $H =< V, E, l >$ where V is a set of nodes representing concepts, E is a set of edges denoting the relationship between concepts, and l is a function from $V \bigcup E$ to a set L of string labels, the text form of the concepts and relationships.

Definition 2. *Document clasification*

A set of documents D is classified in a hierarchy, H, with a function $\mu : V \rightarrow 2^D$. where V are the nodes representing concepts. μ represents the most specific concept v that can be found.

Definition 3. *Mapping two ontologies*

A mapping M, from one ontologies, H, to another H' is a function $M : VxV' \rightarrow rel$ where rel is a set of mappings { $\supseteq, \subseteq, \star, \perp, \equiv$ } .
$v1 \supseteq v2$: v1 is more general than v2.
$v1 \subseteq v2$: v1 is less general than v2.
$v1 \star v2$: v1 is a compatible with v2.
$v1 \perp v2$: v1 is disjoint with v2.
$v1 \equiv v2$: v1 is a synonym of v2.

Definition 4. *Sense Refinement (Elimination Rules)*

Given a label l with senses, $sense(l) = \{l\#1, l\#2, ..., l\#n\}$. sense $l\#i$ can be removed if one of the following holds.

1. Both of the following hold

   for some sense $l\#j$ there is an ancestor l' of l and a sense $l\#k$, such that $l\#jwl\#k$

   there is no ancestor l' of l and no $l\#k$, such that $l\#i \geq l\#k$.

2. Both of the following hold

   for some sense $l\#j$ there is a descendant l' of l and a sense $l'\#k$, such that $l\#j \leq l'\#k$

   there is no descendant l' of l and no $l'\#k$, such that $l\#i \leq l'\#k$.

3. there is a parent l' of c such that for all $l'\#j$, $l'\#j \perp l\#i$

Definition 5. *Sense Refinement (Composition Rules)*

Let l and l' be two labels with senses $l\#i$ and $l'\#j$. We can replace sense $l\#i$ with $l\#i - l'\#j$ if either

1. $l'\#j \leq l\#i$

2. $l\#i \geq l\#j$

2.1 Related Work

As enabling interoperability between heterogeneous systems and ontologies is important for the future of the semantic web, there are many different ways of doing ontology mapping [7]. Ontology mapping is the ability to map a concept from one ontology to another concept in another, which can be accomplished many ways. The main different approaches come in a couple of broad categories: those that are lexicon based, using the textual values of the names to provide information; instance based, utilizing the values inside the context as opposed to the label; machine learning and probability based approaches, which use prior knowledge or training values on which to match labels. Some projects that utilize these concepts are described in slightly more detail here, LSD, MOMIS, and Context OWL.

Learning Source Description(LSD) is a project that creates mappings using a multi-strategy learning approach [9]. This approach uses multiple base learners: the name learner, the content learner, the naive bayes learner, and the xml learner. Each of these learners contributes a score to a final classification. LSD then uses a training and classifying phase to accomplish the mapping. During the training phase small data-sets that have been manually mapped are used as the training. While LSD has a high degree of accuracy the need for a manually curated data set prevents it from being utilized in a larger context.

Another method is called MOMIS(Mediator Environment for Multiple Information Sources) [3]. MOMIS takes all the information sources and creates a global virtual view, a single ontology from all heterogenous sources. This is done through a five phase process where : 1) local schemas are extracted , 2) local source is annotated using WordNet, 3) A common thesaurus is generated using inter and intra schema knowledge, 4) a global view is generated by using the common thesaurus, 5) finally the view is annotated by utilizing the local schemas from step 1. MOMIS allows for automated mapping and utilizes Wordnet as this paper does but doesn't utilize the logical meaning within the schemas.

Context OWL (Contextualizing Ontologies) is a work which utilizes the concept of bridge rules and is the predecessor to this work [5]. Here the authors have utilized the fact that ontologies cannot be merged if they contain inconsistent concepts, but they can if a set of bridge rules can be applied to the concepts. These bridge rules are defined as $\supseteq, \subseteq, \star, \bot, \equiv$. The distinction between this predecessor paper and the more recent is the lack of logical deduction and the lexical analysis that precedes the matching.

2.2 Methods

The algorithm has two phases, semantic explicitation and semantic comparison. An ontology has a lot of information that can be gleaned from labels and the graph structure. Semantic explicitation takes an ontology and adds as much meaning to the nodes and edges and represents them as a logical formula. For all nodes,v, and edges, e, semantic explicitation is a function that where $v \rightarrow w(v)$ and $e \rightarrow w(e)$, where $w \in W$ is a logic. In this paper, W, is an intermediate logic,resembling first order logic, that uses the meta-data gleaned from WordNet, and is used as an intermediary stepping logic that can be converted into another logic, such as a first order logic.

In the semantic comparison phase we find the mappings between two concepts $w(k)$ and $w(k')$ by converting them into a problem of satisfiability inside a logic W', and performing queries. For this paper a basic transformation from $W \rightarrow W'$ is done, where W' is first order logic in conjunctive normal form (CNF). Notice that W and W' could be any logic, and any solver that can accept these logics could be used. For example the Sat4J solver is used in this paper but z3 was heavily considered [4, 8].

2.2.1 Semantic explicitation

The first phase is creating the meta-data for the ontologies, called concept hierarchies inside the paper. The details of this process were found inside a paper by Magnini [12]. This meta-data will be then converted into logical statements in the language of W. First, all labels inside the ontology are run through a script to perform linguistic analysis. For every node label and edge label, l, its part of speech, normal form, the number of senses is discovered along with the hypernyms, meronyms, homonyms, synonyms, and other relationships(currently only subtype of), using WordNet [13]. The part of speech represents where it belongs inside a sentence, verb,noun, etc. The normal form is reducing the word to its base, e.g. trespassing → trespass. Hypernyms are words that are members of the root word, meronyms are words of which the root word is a member, homonyms are words that have the same spelling but different meanings, while synonyms have the same meaning but different spelling. Notice that these word relationships are very similar to logical statements, which gives a rich set of meta-data to use to create rules for these ontologies and make mappings, see definition 3 and example **??**. Pseudo-code for finding the relationships can be found in algorithm 1. An example of a new ontology with meta-data can be seen in figure 1. In the picture you can see the original graph has now been annotated with additional meta-data information that provides extra information about the structure and relationships.

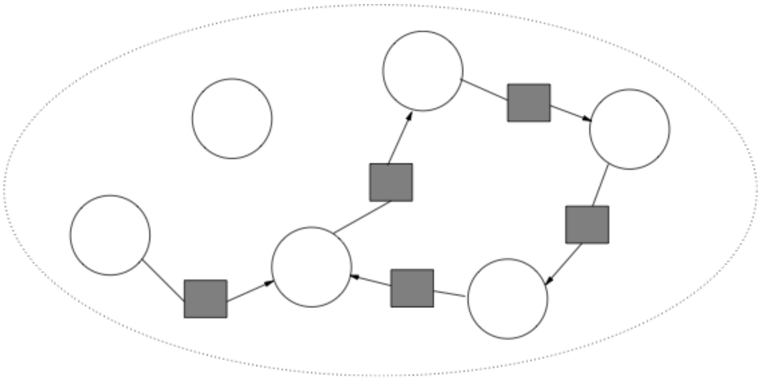


Figure 1: Original ontology, circles, now modified with meta-data

As an example let us look at the word "bodies". Running the word through WordNet shows the following the results. Firstly it recognizes the normal form of "bodies" is "body", it also says that body is a noun which has 11 senses. The first two of these senses are as a physical structure (body#1), or as a social group (body#2), see the example WordNet output **??**.

Once labels have associated meta data then sense refinement is performed. Sense refinement utilizes the graph structure of the ontology to provide more information and either add, or remove current meta-data. As an example let us say that we have the following tuple.

Algorithm 1 Compare Synsets

```
Require: $label_1, label_2$
  $synsets_1 \leftarrow label_1.getSynSets(), synsets_2 \leftarrow label_2.getSynSets()$
  for $s_1$ in $synsets_1$ do
    for $s_2$ in $synsets_2$ do
      if $s_1 == s_2$ then
        return True
      end if
    end for
  end for
  return False
  {This same process is repeated for antonyms, hypernyms, hyponyms. Obviously the code
  shown is naive and there are smarter ways than by rerunning for each relationship type}
```

$< humanbodies, type\_of, anatomy >$. There are three senses of anatomy and none of the senses have a definition as a "social group"(body#2), so we can remove body#2 from the senses list as it does not relate to this ontology. On the other hand anatomy sense #2 is "human body, physical body, ... => physical structure" , so we can keep the body#1 sense. Once a relationship is discovered this is added to a growing list of logic in W. The mapping from WordNet into logical statements is fairly trivial, see figure 2.

For any two labels with senses s, and t, with sense numbers i, and j, inside an ontology.

1. $s\#i \leq t\#j$: $s\#i$ is a hyponym or a meronym of $t\#j$
2. $s\#i \perp t\#j$: $s\#i$ is an antonym of $t\#j$
3. $s\#i \geq t\#j$: $s\#i$ is a hypernym of $t\#j$
4. $s\#i \equiv t\#j$: $s\#i$ is a synonym $t\#j$

This approach of refining labels narrows down a broad labels, such as "bodies", into a particular usage of that word, "body#2" relating with anatomy. Unfortunately, in practice, the number of sense refinements that can be made on an ontology turns out to be small so just the refinements are not enough to capture the original ontology. This is probably due to WordNet still being a work in progress and still not capturing many relationships. It is unclear how the original authors overcame this lack of refinement, possibly by manually extension.

To overcome the lack of refinement, instead of using particular word senses, if a refinement is found it is instead applied to the label. This allows additional logical rules to be applied to the original ontology at the expense not being able to cull extraneous information.

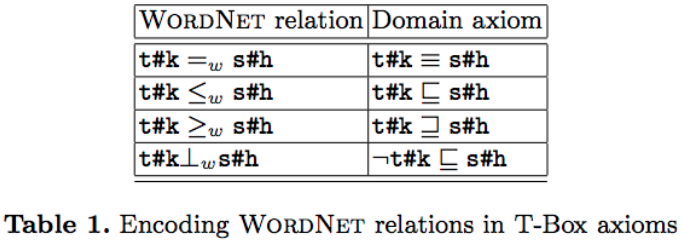
	$Human$	$Bodies$
ID	0	1
$Token$	$Human$	$Bodies$
$normalform$	$human$	$body$
PoS	$noun$	$noun$
$senses$	$\{human\#1, human\#2, ...\}$	$\{body\#1, body\#2...\}$

2.2.2 Mapping

Once semantic explicitation is done on each ontology we have sufficient information to start drawing relationships between the two. To map Ontology 1, O_1, to Ontology 2, O_2 every $w(i)$ is compared to $w(j)$, $i \in O_1, j \in O_2$. The comparison is the same as sense refinement, where all relationships, hypernym, hyponym, etc, are discovered and added to the logical rules, in logic W.

2.2.3 Semantic comparison

The second phase in the process is converting the ontologies, with the meta-data gained in phase 1, in logic W, into logical statements in W'. In this case W' must be some form that can be handled by a SAT solver [11]. In this particular case the SAT solver being used is Sat4J [4], which takes input as CNF, which can be done by any means. In this instance, since the intermediate logic W', does not contain any existential quantifiers this is a fairly simple process.



WordNet relation	Domain axiom
$\mathtt{t\#k} =_w \mathtt{s\#h}$	$\mathtt{t\#k} \equiv \mathtt{s\#h}$
$\mathtt{t\#k} \leq_w \mathtt{s\#h}$	$\mathtt{t\#k} \sqsubseteq \mathtt{s\#h}$
$\mathtt{t\#k} \geq_w \mathtt{s\#h}$	$\mathtt{t\#k} \sqsupseteq \mathtt{s\#h}$
$\mathtt{t\#k} \perp_w \mathtt{s\#h}$	$\neg\mathtt{t\#k} \sqsubseteq \mathtt{s\#h}$

Table 1. Encoding WordNet relations in T-Box axioms

Figure 2: Encoding the wordnet relationships into T-Box axioms.

Now logical queries can be performed between the ontologies. Using figure 3 as an example the problem of checking whether ”Chat and Forum” in google is less general than ”Chat and Forum” in Yahoo becomes the following satisfiability formula.
”Chat and Forum” in Google, assuming each has only one sense, becomes $(assert(\vee chat\#1 forum\#1))$.
This is a subtype of ”Literature#1” in their ontology, $(assert(subtype(\vee chat\#1 forum\#1)literature\#1$.

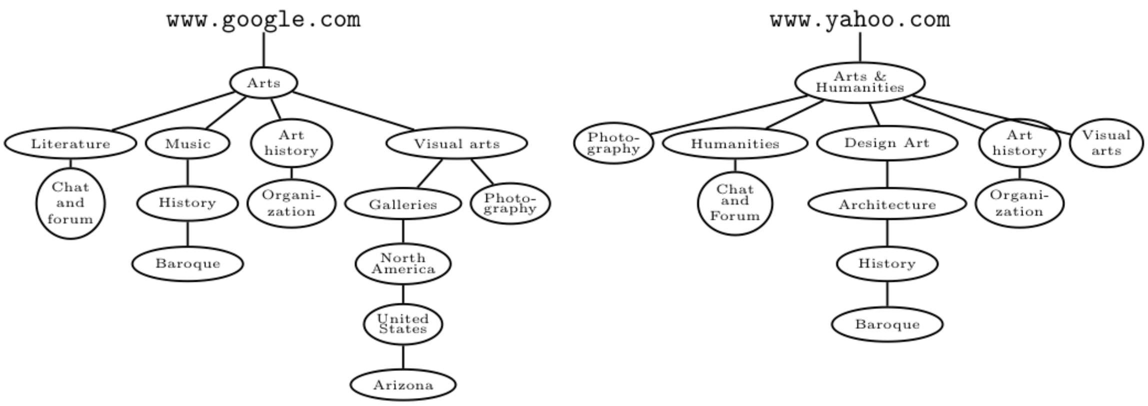


Figure 3: Example ontologies (where each edge is of the form ”is_a”, from Bouquet paper.

expressed as T-Box.

$$\begin{aligned}
art\#1 \sqsubseteq humanities\#1 \\
humanities\#1 \sqsupseteq literature\#2 \\
(art\#1 \sqcap literature\#2 \sqcap (chat\#1 \sqcup forum\#1)) \\
(art\#1 \sqcup humanities\#1) \sqcap humanities\#1 \sqcap (chat\#1 \sqcup forum\#1)
\end{aligned}$$

In this case the solver return satisfiable showing that googles term is indeed less general.

For each concept, k, it is compared to another concept k' for each potential relationship. Equivalence is checked by comparing $w(k) \sqsubseteq w(k')$ and $w(k) \sqsupseteq w(k')$, which also provides whether one concept is more or less general than another. Compatibility is checked by $w(k) \sqcup w(k')$ is satisfiable, while disjoint is checked by verifying that $w(k) \sqcup w(k')$ is not satisfiable.

Summarizing, each node and edge from each ontology, v, is converted into meta-data $w(v)$, where w is a set of WordNet functions and then printed in an intermediate logic W. Afterwards a mapping is performed by comparing every piece of meta-data from each ontology against each other and printed in W. Lastly W is converted to some other logical form that can be handled by a SAT solver, W'.

2.3 Test Sets

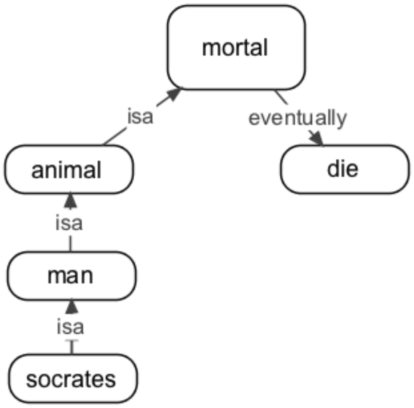
Google web directory: The google web directory is a set of links that relates information by hyperlinks, or as google says ”The web organized by topic into categories”. This can be conceptualized to be an ontology where the concepts are the word, and the relationships are all ”is_part_of”. From this ontology the interrelationship of the links can be matched to itself using the solver. The google web directory can be found here `http://www.google.com/dirhp`.

There are two ontology sets, Garts, and Gbusiness, that denote the google arts directory and the google business directory respectively.

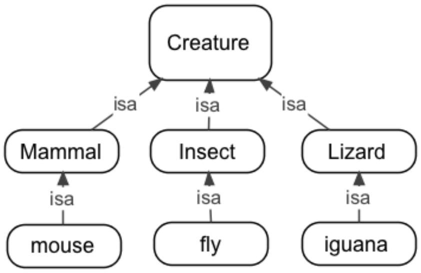
Yahoo web directory: The same as description as the Google web directory. It can be found here `http://dir.yahoo.com/`.
There are two ontology sets, Yarts, and Ybusiness, that denote the yahoo arts directory and the yahoo business directory respectively.

Mortal: A very small hand made ontology set that was used for logical verification. The ontologoies have no words in common besides edge names so that any logical inferences made between them will be strictly the result of a new correctly merged ontology.
The first ontology, mortal1, is a prototypical test philosophical ontology containing,see figure 4(a).
$< socrates, is_a, man > < man, is_a, animal > < animal, is_a, mortal > < mortal, eventually, die >$

The second,mortal2, is a small animal hierarchy, see figure 4(b).
$< mammal, is_a, creature > < insect, is_a, creature > < lizard, is_a, creature > < mouse, is_a, mammal >$
$< fly, is_a, insect > < iguana, is_a, lizard >$



(a) Mortal 1 Ontology



(b) Mortal 2 Ontology

Table 1: Datasets and their size and meaning

Name	# of Triples	Expresses
Garts	147	Arts & Music
Yarts	33	Arts & Music
Gbusiness	79	Business
Ybusiness	124	Business
mortal1	4	Mortality
mortal2	6	Creatures

3 Experimental Results & Discussion

Several different tests were run to test varying parts of the algorithm. The first test is discovering how well the sense refinement worked. This tested how much additional information could be gleaned from the ontology structure that was not originally included inside the ontology. Ontologies are often created by hand and this refinement has the possibility of fleshing out the ontology automatically.
The second test shows how well two ontologies become mapped together. Being able to map two ontologies together is important for knowledge integration.
The final test is querying the resultant merged ontology. This is a verification of the entire process.

3.1 Sense Refinement

To test sense refinement the first several levels of the Google web directory and Yahoo web directory, under the Business and Arts sections, were analyzed along with my example ontologies, see table 4. As can be seen in table 4 the number of refinements that can be done on any one particular ontology is not much. This was the reason I had to abandon individual senses. This also means that as an ontology refinement tool this has potential but as it stands would not be significant. The sense refinement also has the potential for incorrect relationships, as can be seen in table **??** The relationships were considered to be incorrect by manual verification. For example the following logic was found $healthcare \equiv business$, obviously not correct. On the other hand some additional rules were found such as $agribusiness \equiv agriculture$.

As can be seen the false positive rate for sense refinement is high, and while while the sample size is small, and seems to range between 20% and 50%. For any sort of practical use this number would have to be reduced significantly.

Table 2: # of Sense refinements

Name	# of Relationships	# Refinements	# Incorrect
Garts	147	4	2
Yarts	33	0	0
Gbusiness	79	3	2
Ybusiness	124	10	3
mortal1	4	0	0
mortal2	6	0	0

3.2 Mapping

For the mapping testing similar ontologies were merged together using the logical mappings. Afterwards the number of new relationships were counted. The number of relationships were too many and too complicated to be manually verified, so false positives were not counted. For correctness the merged ontology logic was run through the sat solver, if the new ontology was satisfiable then it was considered to then at least it was logically consistent. Of the merged ontology logics only the Business merge was unsatisfiable. This means that some logical inferences were incorrect, though is was unclear which logical statement caused the error. Most likely some predicate became a superset or a subset through the word matching that with greater interpretation would not have been created.

When it does work the results are very pleasing. For example in the Mortal ontology the following new relationships were found.

$$\begin{aligned} animal &\equiv creature \\ mortal &\sqsupseteq mouse \\ mortal &\sqsupseteq creature \\ die &\sqsupseteq fly \end{aligned}$$

Making the merged ontology the following, see figure 4.

The most important is the process correctly identifies that a creature is the same as an animal. This means that despite the ontologies having no terms in common, the new merged ontology can now speak to the mortality of flies, mice, and iguanas!

3.3 Querying the New Mapped Ontology

As a final verification some logical queries were run on the Arts and Mortal ontologies, Business queries were not run as it was inconsistent. As expected as long as the query is formed correctly the satisfiability was correct.

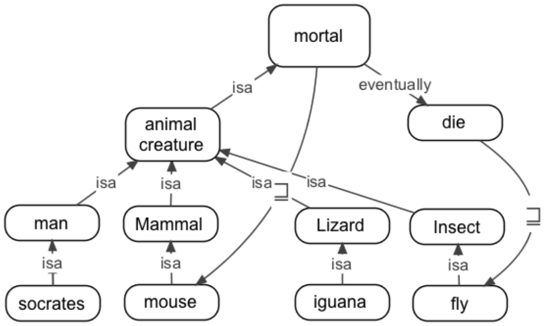


Figure 4: Merged Mortal Ontology

Table 3: Mapping Ontologies

Name	# of Relationships	# New Relationships	Time	Consistent
Arts	180	95	2.4 sec	Yes
Business	203	552	6.7 sec	No
Mortal	10	4	0.1 sec	Yes

As two examples:

1. Is a mouse mortal? query triple, $< mouse, not_a, mortal >$. result UNSATISFIABLE
2. Is indie rock really a type of art? query triple, $< indie, not_a, art >$. result UNSATISFIABLE

So despite the concept of mortality being in a separate ontology as mouse, and despite the concept of music as art being in a different ontology than indie rock, both were shown to have been returned correctly. The converse is also true, changing the two example queries from not_a to is_a produce SATISFIABLE.

Table 4: Mapping Ontologies

Name	# Queries	# Correct
Arts	5	5
Mortal	3	3

3.4 Drawbacks

There are a couple of flaws in this algorithm. The first is that it attempts to find the relationships between the ontologies by doing a pairwise calculation for all labels. While this approach works, with moderately large ontologies, see 4, there is a chance that it will become logically inconsistent. This is due to a false positive matching between terms that creates a logical fallacy. If every term is matched against all other this seems like an inevitability. A potentially better approach would be to only match labels that were ”close” to each other, perhaps by only matching labels that are within a couple of hops from each other.

Another flaw in the mapping is that instance information is not used. The structure of the schema and the labels provides a good deal of information that can be used in a mapping, but there are many cases where this is not sufficient. For example, what if instead of the label ”States” the schema creator had used ”ST_CODE” (state code). One could imagine a system that could also look through the instance information, in this case it would contain such things as ”TX”, ”AK”, ”NV”, ”NY”, etc, and be able to deduce that in fact ”ST_CODE” is equivalent to states. Once the additional information is known then a more complete mapping could be accomplished.

4 Conclusion

The internet is a global spaced that is filled with information. This information comes from interlinked documents and is normally backed by databases, with database schemas. Merging this information together to provide machine queryable data is the idea behind the Semantic Web. The common interface for data inside the semantic web is the Resource Description Framework (RDF), which is a triple of information. These triples when combined together form ontologies, a set of interlinked data representing concepts and relationships.

A major hurdle in creating the semantic web is semantic interoperability. Semantic interoperability is a concept that describes the fact that ontologies are created by differing entities and rarely are the labels that they use to describe the same concepts and relationships the same. This project tries to overcome this hurdle by using a combination of a common vocabulary and SAT solving to create the relationships between the ontologies.

The results from the completed algorithm are a mixture of both positive and negative results. On the negative side: the sense refinement on a single ontology provides some additional information, but not much and the propensity for a false positive is high, 20%-30%. Inside the ontology mapping there is also the chance that when ontologies become too large the mapping will be logically inconsistent, as happen in the Business ontology merge. The positive results show that the ontology mapping does work and can provide a merged ontology that contains more information than either ontology separately. This is shown in both the Arts and Mortal ontologies where new queries can be formed that would be impossible to check on either ontology separately, but on the merged ontology are shown to be verifiable.

5 Practical Details

To actually run the algorithm all the necessary files are located here at `http://www.cs.utexas.edu/~parnell/projects/formver_files/`, this includes a readme and details. Inside the url you will find the following.

1. phase1.py : Script I wrote to implement phase 1 of the algorithm. Takes in files in a triple format and converts them into a intermediate logic format after sense refinement and mapping.
2. phase2.py : Script I wrote to implement phase 2 of the algorithm. Takes in a file in intermediate logic format and converts it into CNF format.
3. org.sat4j.core.jar : Jar file of Sat4J. Written by the good people at `http://www.sat4j.org/`, [4].
4. Six Ontologies : All six of the ontologies used in this paper, the Business, Arts, and Mortal.
5. Readme File : Some guidance on how to run the programs

References

[1] M. Ashburner, C. A. Ball, J. A. Blake, D. Botstein, H. Butler, J. M. Cherry, A. P. Davis, K. Dolinski, S. S. Dwight, J. T. Eppig, M. A. Harris, D. P. Hill, L. Issel-Tarver, A. Kasarskis, S. Lewis, J. C. Matese, J. E. Richardson, M. Ringwald, G. M. Rubin, and G. Sherlock. Gene ontology: tool for the unification of biology. the gene ontology consortium. *Nature genetics*, 25(1):25–29, May 2000.

[2] Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. DBpedia: a nucleus for a web of open data. In *Proceedings of the 6th international The semantic web and 2nd Asian conference on Asian semantic web conference*, ISWC'07/ASWC'07, pages 722–735, Berlin, Heidelberg, 2007. Springer-Verlag.

[3] D. Beneventano, S. Bergamaschi, F. Guerra, and M. Vincini. Synthesizing an integrated ontology. *Internet Computing, IEEE*, 7(5):42 – 51, sept.-oct. 2003.

[4] Daniel Le Berre and Anne Parrain. The sat4j library, release 2.2. In *JSAT*, 2010.

[5] Paolo Bouquet, Paolo Bouquet, Fausto Giunchiglia, Frank Van Harmelen, Luciano Serafini, and Heiner" Stuckenschmidt. C-owl: Contextualizing ontologies. *Journal of Web Semantics*, 1:164–179, 2003.

[6] Paolo Bouquet, Bernardo Magnini, Luciano Serafini, and Stefano Zanobini. A SAT-based algorithm for context matching. In Patrick Blackburn, Chiara Ghidini, Roy Turner, and Fausto Giunchiglia, editors, *Modeling and Using Context*, volume 2680 of *Lecture Notes*

in Computer Science, chapter 6, pages 66–79. Springer Berlin / Heidelberg, Berlin, Heidelberg, June 2003.

[7] Namyoun Choi, Il-Yeol Song, and Hyoil Han. A survey on ontology mapping. *SIGMOD Rec.*, 35:34–41, September 2006.

[8] Leonardo de Moura and Nikolaj Bjørner. *Z3: An Efficient SMT Solver*, volume 4963/2008 of *Lecture Notes in Computer Science*, chapter 24, pages 337–340. Springer Berlin, Berlin, Heidelberg, April 2008.

[9] AnHai Doan, Pedro Domingos, AnHai Doan, Pedro Domingos, and Alon Y. Levy. Learning source description for data integration. In *WebDB (Informal Proceedings)*, pages 81–86, 2000.

[10] Thomas R. Gruber. A translation approach to portable ontology specifications. *Knowl. Acquis.*, 5(2):199–220, June 1993.

[11] B. M. Magnini, L. Serafini, A. Dona, L. Gatti, and M. Speranza. Large-scale evaluation of context matching. Technical report, Istituto Trentino di Cultura, January 2003.

[12] Bernardo Magnini, Luciano Serafini, and Manuela Speranza. Linguistic based matching of local ontologies. In *Working Notes of the AAAI-02 workshop on Meaning Negotiation. Edmonton*, 2002.

[13] Michael M. Stark and Richard F. Riesenfeld. WordNet: An electronic lexical database. In *Proceedings of 11th Eurographics Workshop on Rendering*, 1998.