Scaling Alignment of Large Ontologies

Suzette K. Stoutenburg and Jugal Kalita
School of Engineering and Applied Science
University of Colorado at Colorado Springs
Colorado Springs, Colorado 80918
Email: suzette@coloradostoutenburg.com Email: kalita@eas.ucsc.edu

Kaily Ewing and Lisa Hines
The College of Letters, Arts and Sciences
University of Colorado at Colorado Springs
Colorado Springs, Colorado 80918
Email: kewing@ucsc.edu Email: lhines@ucsc.edu

Abstract—In recent years, the number of ontologies shared on the Web has increased dramatically, supporting a growing set of applications such as biological knowledge sharing, enhanced search and discovery, and decision support. This proliferation of Web knowledge sources is resulting in an increasing need for integration and enrichment of these knowledge sources. Automated solutions to aligning ontologies are emerging that address this growing need with promising results. However, only very recently, solutions for scalability of ontology alignment have begun to emerge. The goal of this research is to investigate scalability issues in alignment of large-scale ontologies. We present an alignment algorithm that bounds processing by selecting optimal subtrees to align and show that this improves efficiency without significant reduction in precision. We apply the algorithm in conjunction with our approach that includes modeling ontology alignment in a Support Vector Machine.

Index Terms—Knowledge Acquisition, Knowledge Management, Machine Learning, Ontology, Ontology Alignment, Semantic Technology, Support Vector Machines.

I. INTRODUCTION

With the advent of the Semantic Web, standard methods for defining the semantics of terms are allowing concept sharing on a wide scale. Ontologies represented in a standard language, such as OWL, and exposed on a network offer the potential for unprecedented interoperability solutions since they are semantically rich, machine-interpretable and inherently extensible. The number of ontologies available on the Web has increased significantly in recent years, particularly in the Biomedical domain; for example, the Open Biomedical Ontologies [24] and the Gene Ontology [12]. However, as different parties generate ontologies independently, the level of heterogeneity across platforms increases [6]. And, despite use of a standard language for ontology representation, such as OWL, there are still vast differences in ontology design. Therefore, there is a growing need to integrate ontologies using automated methods.

To reduce the impediments to integrating data sources using ontologies, researchers are seeking to apply ontology alignment techniques to automatically discover relationships across ontological components. Ontology alignment (also referred to as matching [6], [21], mapping [19], and merging [18]), is the process of discovering relationships between entities across two or more ontologies based upon a variety of techniques that use entity labels, structure, semantics, and external resources to determine the relationships. Schema matching is also a related term that refers to earlier work in matching, primarily over databases [3]. The differences between schema integration and ontology alignment are well defined in [19]. We will use the term ontology alignment in this work. Ontology alignment is viewed as a potential solution to a broad set of challenges, including data integration, agent negotiation, and Web service composition [4] as well as peer-to-peer information sharing and advanced navigation and query answering on the Web [6]. There has been success in applying ontologies for integration in biotechnology [22], [31], [5], [29] and in prototypes for interoperability in defense applications [14], [26].

The original goal of this research was to apply a Support Vector Machine (SVM) approach to align large-scale biomedical ontologies using non equivalence relations. Details of this approach can be found in [25]. At the onset of large-scale testing, however, we encountered significant challenges in opening large-scale ontologies for processing. We found similar challenges even when using open source tools that are widely used in the semantic community, such as the Protege tool for editing and managing ontologies. In addition, when feature extraction for the SVM was applied in the alignment process, the application ran initially for up to three days when processing fairly small numbers of ontology pairs. Therefore, it became apparent that investigating scalability would be an essential part of this research. Our work in this area is the focus of this paper.

II. RELATED WORK

In 2009, scalability of ontology alignment remains a key challenge and approaches to address the challenge are only starting to emerge. In the Ontology Alignment Evaluation Initiative (OAEI) in 2007 and 2008, participants were encouraged to align large data sets, including the Anatomy, Food and Library ontologies [10], [1]. However, only four of seventeen tools accomplished alignment tasks with the largest ontologies in OAEI 2007 and only three in OAEI 2008. Paulheim found that the very large biomedical ontologies, such as GO and NCI Thesaurus cannot be processed by top performing ontology alignment tools [20]. In addition, the Harmony Integration

1By “large” ontologies, we mean ontologies with thousands of classes, as in the Gene Ontology which has more than 25,000 classes.

2http://oaei.ontologymatching.org/2008/

3http://ncicb.nci.nih.gov/NCICB/infrastructure/cacoreOverview/vocabulary
Workbench authors indicate that it is only after five years of development that they are starting to address the challenge of scalability in alignment [16]. This is further evidence of the difficulty of the challenge.

To date, research into developing scalable infrastructures for processing and reasoning over ontologies has been in two main areas: first, developing a representation approach that combines semantic richness (i.e., expressiveness) with scalable performance; and second, harnessing the power and scalability of database systems to represent ontologies [8]. Solutions to scaling ontology alignment are only starting to emerge. In fact, scalability continues to be one of the grand challenges of ontology construction and use [8] and scaling ontology alignment in particular remains a research area [7].

Some early attempts at scaling ontology alignment include initial elimination of candidates prior to alignment. Typically, light-weight string-based matching has been applied to handle the task of candidate elimination in large-scale ontology alignment; for example, [32], [11]. The problem with these approaches is that they are comparing lexical qualities of ontological classes instead of semantic qualities. Thus, this approach risks reduced accuracy, in particular, reduced recall. Some of the more successful approaches that use string-based matching to eliminate candidates perform alignment using domain-specific knowledge sources [32], [11]. Use of domain-specific knowledge sources, however, precludes these solutions from performing successful ontology alignment in other domains. Other work has adapted match algorithms to reduce the use of memory. Mork et al. showed how two large biomedical ontologies, the Foundational Model of Anatomy (FMA) and Galen\(^4\), could be aligned by considering only direct subclasses and superclasses of each class to align. This work was successful in aligning large biomedical ontologies, but it is important to note that the ontologies were not expressed in OWL nor RDF; instead a frame-based system was used [15].

As discussed earlier in this section, very few alignment approaches have addressed the challenges of large scale ontology alignment. In OAEI 2007, only four of seventeen approaches aligned the largest set of ontologies, which include the Anatomy, Food and Library tracks [10]. One of the approaches, DSSIM, applied manual partitioning in order to process the large ontologies [17]. Other tools, including RIMOM [28] and PRIOR+ [13] applied light-weight string-matching techniques as alternatives to full-scale alignment. An approach by Wang and Xu was presented at OAEI 2007 called Lily, but Lily did not align the largest ontologies until OAEI 2008. Wang and Xu make reference to an approach to scalability which they refer to as “generic ontology matching method based on semantic subgraphs,” but this work has not yet been published [30]. At the Ontology Matching workshop in 2008, Paulheim proposed two approaches to enhance the scalability of ontology alignment: overlapping partitions and optimized thresholding [20]. These are discussed in further detail in the next section.

Partitioning ontologies as an approach to improving scalability is risky, since most ontology alignment approaches rely on ontological structure to some degree. Performed naively, partitioning could lead to significant degradation in alignment accuracy. One approach to avoiding this issue involves the creation of overlapping partitions. In this approach, the direct neighbors of each concept are included in the partition, thus creating overlapping partitions. This approach has shown significant improvement in alignment accuracy though there is significant cost in time complexity [20].

A divide-and-conquer strategy for ontology alignment is described in work by Hu in late 2008 [9]. In this approach, ontologies are partitioned into blocks by measuring structural proximity, with each block being labeled with RDF sentences [33]\(^5\). Then, each block is mapped using entities matched beforehand and alignments are determined across blocks [9]. Hu reports that this approach reduced execution time of alignment, down to 12 minutes for the OAEI Anatomy track. By comparison, in OAEI 2008, DSSIM, RIMOM and PRIOR+ report execution times of 4 hours, 75 minutes and 23 minutes, respectively. However, the execution time for the partitioning elements of the algorithm is not reported; in addition, the partitioning algorithm uses a structural proximity matrix that is precomputed. Based on our research results [25], we believe this is likely because the partitioning process likely took days to run even on moderately sized ontologies, containing no more than 3500 classes each. Furthermore, Hu’s structural proximity matrix does not consider the qualitative properties of the relations between ontological nodes in the graph. That is, each link in the graph is treated as if were the same relation. Thus, if the relation between ontological concepts is disjoint, or some other relation that indicates significant difference, then the structural proximity matrix approach breaks down in ontology alignment.

Stuckenschmidt and Klein propose an algorithm for partitioning ontologies for enhanced visualization and navigation [27]. This approach measures the degree of dependency between concepts, but does not maintain semantics of the partitions [2] Algorithms are proposed in [18] and [23] to partition ontologies by traversing semantically close links (i.e., relationships between concepts) and produce stand-alone fragments. However, as pointed out in [2], these approaches do not characterize the logical properties of the extracted fragments. The notion of ontology modules and methods for extracting them are formally defined in [2]. These efforts partition ontologies based on the semantics of the terms. However, despite the strong accuracy of the results in [2] for example, these partitioning methods often result in modules (or partitions) that are still quite large, on the order of thousands of classes [9].

A second approach to enhancing scalability is the idea of thresholding. In this approach, a lower threshold to select

\(^4\)http://www.co-ode.org/galen/

\(^5\)An RDF Sentence is a unit of summarization used to summarize the content of an ontology [33].
pairs for alignment is used across partitioned ontologies. This approach was motivated by the observation that average confidence values for alignment are significantly lower for false positives than for true positives. Threshold levels were obtained by manually inspecting the optimal values across a set of data. It is claimed that this approach improves accuracy without significant increased cost in time complexity, over other partitioning methods [20].

III. Definitions

In this section, we define ontology, ontology alignment, and the relationships used to align ontologies in this work.

A. Ontology

There are many formal definitions of ontology that have been published. Typically, definitions are designed to meet the needs and goals of the researcher but there are some core components that comprise any definition. At a minimum, an ontology can be defined to consist of a nonempty set of classes, a nonempty set of properties or relations, a hierarchy among classes, a hierarchy among properties, and a set of functions that map classes via properties, the latter being an expression of class relationships. Ehrig defines this set of features as Core Ontology [4]. Some definitions of ontology include instances [6] while others define instances and the instantiation function to be part of a Knowledge Base [4]. Interestingly, Euzenat explicitly defines a subset of relations that exist in an ontology, such as subclass (he calls it specialization), disjointness (exclusion), instantiation and assignment [6]. Euzenat also defines data types to be part of an ontology while Ehrig defines data types to be a special case of class. Also, Ehrig states that an ontology must consist of a lexicon that defines the names of classes and relations and a lexical reference for those names. We prefer Ehrig’s approach to defining ontology since it provides a modular definition of the components of an ontology in such a way that operations and algorithms can be more precisely defined. However, we use some elements of both approaches in the definition below.

Definition 1 Ontology

An ontology is a 6-tuple \( O = < C, R, \leq_C, \leq_R, \sigma, A > \), such that:

- \( C \) is a nonempty set of classes: \( \{ C_1, C_2, ... C_n \} \),
- \( R \) is a nonempty set of relations: \( \{ R_1, R_2, ... R_n \} \),
- \( C, R \) are disjoint,
- \( \leq_C \) is a class hierarchy, a partial order on \( C \),
- \( \leq_R \) is a relationship hierarchy, a partial order on \( R \),
- \( \sigma : R \rightarrow C \times C \), representing relationships between classes,
- \( A \) is a set of class axioms, possibly empty: \( \{ A_1, A_2, ... A_n \} \).

Note that in this work, we do not plan to operate over instance data. Therefore, we leave instances out of the definition of ontology. Also, the term property should be considered synonymous with relation.

Similar to Ehrig’s definition, we denote that if \( c_1 \leq_C c_2 \), where \( c_1, c_2 \in C \), then \( c_1 \) is a subclass of \( c_2 \) and \( c_2 \) is a superclass of \( c_1 \). If \( c_3 \leq_C c_2 \) and \( c_4 \in C \) such that \( c_1 \leq_C c_3 \leq_C c_2 \), then \( c_1 \) is a direct subclass of \( c_2 \) and \( c_2 \) is a direct superclass of \( c_1 \). We also borrow, in part from Ehrig, the concept of a set of entities and we say that an entity \( e \in E \) in ontology \( O \) is a class or relation, such that \( e \in O \in C \cup R \). We treat data types as a set \( D \subseteq C \), i.e., a special set of classes, as in [6].

B. Ontology Alignment

To formally define ontology alignment, we first define correspondence, inspired by [4]. This definition uses the same sets \( C, R, \) and \( E \) defined above.

Definition 2 Correspondence

Consider two ontologies \( o_i, o_j \). Let \( C_i \) and \( R_i \) represent classes and relations in \( o_i \), respectively. Let \( C_j \) and \( R_j \) represent classes in \( o_j \), respectively. A correspondence \( \text{CORR} \) is a 4-tuple \( = < e_i, e_j, \phi, p > \), where:

- \( e_i \in o_i, e_j \in o_j \),
- if \( e_i \in C_i \), then \( e_j \in C_j \), or if \( e_i \in R_i \) then \( e_j \in R_j \),
- \( \phi \) represents a nonempty set of relationships, and,
- \( \leq_C \) is a class hierarchy, a partial order on \( C \),
- \( p \in R \) such that \( p \in [0, 1] \), denoting a confidence on the relationship \( \phi \).

We define \( \phi \) as the set of relationships that will be acquired using ontology alignment techniques. These relations may not exist in the ontologies. For example, most ontology alignment techniques seek to acquire equivalence across ontological entities. In this work, we seek to acquire hyponymy. However, as discussed in Section 1.3, our work is distinguished from previous work in ontology alignment because we also seek to discover what we call generic relation \( r \), that is, we seek to acquire relations between ontological entities \( e_i, e_j \) that exist in at least one of the ontologies. This is shown in Figure ?? and explained further in the next Section.

Given this definition of correspondence, we can now define an ontology alignment, which is very similar to [4].

Definition 3 Ontology Alignment

An ontology alignment is a set of correspondences.

C. Relations used to align ontologies

One of the primary goals of this research is to acquire relationships between ontological entities beyond similarity and equivalence. As discussed in Section 2, we define the set of relationships \( \phi \) as a special set of relationships between entities that our algorithms are designed to discover. These relationships include the following.

- Hyponymy, a relation that denotes a subcategory of a more general class; can also be thought of as subclass.
• Hypernymy, a relation that denotes a generalization of a more specific class; also can be thought of as a superclass.
• Correspondences such that \( \phi \in R_1 \cup R_2 \); that is, we will discover properties across ontological entities that are defined in the original ontologies, as depicted in Figure 1. We will define these set of relations to be in the set General Relation \( r \).

IV. DOMAIN FOR ALIGNMENT

In this work, we performed ontology alignment in the Biomedical domain. We chose to align components of the Gene Ontology (GO) [12] with the Mammalian Phenotype (MP) ontology [24], part of the Open Biomedical Ontologies (OBO). The Gene Ontology provides a controlled vocabulary to describe gene and gene product attributes in any organism [12]. It consists of three major categories of descriptors: biological processes, molecular function and cellular component. The Mammalian Phenotype Ontology supports annotation of mammalian phenotypes in the context of mutations, quantitative trait loci and strains that are used as models of human biology and disease [24]. The MP ontology is primarily based on experimental studies conducted using mouse as the model organism. We chose to align these two ontologies because: 1) they are well-developed and frequently updated, 2) they are widely used by the biomedical research community, and 3) they have great potential for application in the area of translational research. For example, discoveries in experiments conducted with mice could then be linked to various gene products that may share similar GO attributes, which may ultimately contribute to hypothesis generation with regard to human disease development. Details on the ontologies used in this evaluation are provided in Table I.

V. ONTOLOGY ALIGNMENT APPROACH

We modeled ontology alignment in a Support Vector Machine (SVM). Consider a set of ontologies we collect from the Web: \( O = o_1, o_2, \ldots, o_n \). We operate over one ontology pair at a time. Let an ontology pair be \((o_i, o_j)\). Let the set of \( k \) classes in ontology \( o_i \) be denoted by \( c_{i1}, c_{i2}, \ldots c_{ik} \). Let the set of \( l \) classes in ontology \( o_j \) be denoted by \( c_{j1}, c_{j2}, \ldots, c_{jl} \). Let \( R \) be the set of non equivalence relations to acquire cross-ontology; that is, \( R = \{ \text{subclass}, \text{superclass}, \text{inferred relations} \} \). Ontology alignment is the process of evaluating each class pair \((c_{ik}, c_{jl})\) to see if a relationship in \( R \) exists between them. We model this process in an SVM by letting each object in the SVM represent a class pair. We model a set of features that exist between each class pair in order to evaluate whether \( R \) exists between the pair.

The architecture used in this work for ontology alignment is shown in Figure 1. The architecture consists of a Feature Extractor and Support Vector Machine. The Feature Extractor identifies the features that exist between each class pair using the feature types described in [25]. The Feature Extractor applies multiple techniques, including string-, linguistic-, structure-, semantic- and pattern-based approaches to derive features. In addition, the Feature Extractor makes use of the semantics of OWL and consults external knowledge sources including OpenCyc and WordNet. The Feature Extractor represents each class pair as a vector of features identified. The SVM utilizes training data to optimize parameters and to construct a model. The SVM model is used to predict class membership between each class pair; that is, the SVM predicts whether a particular relationship exists between each class pair based on the features extracted for each pair. Of course, a unique Feature Extractor and SVM model were developed for each relationship to be acquired cross-ontology. The alignment result is expressed as an XML 4-tuple, including each class from the class pair, the relationship, and a relationship indicator indicating whether the relationship exists between the pair. In this work, we predicted discrete values, so the relationship indicator is \( \in [0, 1] \).

VI. METRICS FOR LARGE-SCALE ONTOLOGY ALIGNMENT

Before we present our Branch and Bound algorithm, we must first discuss the metrics we used to evaluate the accuracy of our approach. Determining precision and recall in large-scale ontology alignment is difficult. Ideally, we would have access to a complete reference alignment with all “true” results; but to build such a set of data for large-scale alignments is manual, difficult and time-consuming, requiring large groups of domain experts. This is particularly true in alignments with non equivalence relations. Data sets with known results for alignment with subclass, superclass and inferred relations simply do not yet exist. Therefore, it is necessary to use metrics that estimate precision and recall. We decided to use a set of metrics proposed by [11] which applies rough approximations for precision and recall using relative quality measures. Kirsten identifies two primary accuracy approximation metrics: match coverage, designed to approximate recall, and match ratio, designed to approximate precision. Match coverage essentially measures how many classes are identified in the alignment. Let \( c_{o1} \) be the set of all classes in ontology \( o_1 \) and let \( \ell_{o1-match} \) be the set of concepts in \( o_1 \) that appear in any alignment pair. Let \( c_{o2} \) be the set of all classes in ontology \( o_2 \) and let \( \ell_{o2-match} \) be the set of concepts in \( o_2 \) that appear in any alignment pair.
TABLE I
BIOMEDICAL ONTOLOGIES USED IN PERFORMANCE EVALUATION. NOTE: THE COLUMN DENOTED WITH † IS USED TO INDICATE THE SHORT NAME OF THE ONTOLOGY, FOR REFERENCE THROUGHOUT THIS DOCUMENT.

<table>
<thead>
<tr>
<th>†</th>
<th>Ontology</th>
<th>OWL Classes</th>
<th>OWL Properties</th>
</tr>
</thead>
<tbody>
<tr>
<td>go</td>
<td>Gene Ontology <a href="http://www.geneontology.org/">http://www.geneontology.org/</a></td>
<td>26763</td>
<td>4</td>
</tr>
<tr>
<td>mp</td>
<td>Mammalian Phenotype Ontology <a href="http://www.informatics.jax.org/searches/MP_form.shtml">http://www.informatics.jax.org/searches/MP_form.shtml</a></td>
<td>29205</td>
<td>1</td>
</tr>
</tbody>
</table>

Then,

\[
\text{MatchCoverage}_{o_1} = \frac{|c_1\text{-match}|}{|c_{o_1}|}
\]

\[
\text{MatchCoverage}_{o_2} = \frac{|c_2\text{-match}|}{|c_{o_2}|}
\]

According to Kirsten, high match coverage indicates that a significant number of concepts were matched. For example, 90% match coverage would indicate that 90% of the classes are matched, which would suggest high recall. Therefore, in general, Kirsten recommends attaining match coverages that are high, above 60%. Of course, this metric assumes that it is expected that a large number of concepts should match across ontologies. This may not always be the case, so low match coverage measures may not necessarily be an indicator of poor recall.

To estimate precision, match ratio measures the ratio between the number of correspondences and the number of matched concepts. The idea is that precision of a matched result is better if a single concept is matched to fewer concepts (preferably similar concepts) and not loosely matched [11]. To define match ratio, let \(\text{corr}_{o_1,o_2}\) be the correspondences found in the alignment process. Then,

\[
\text{MatchRatio}_{o_1} = \frac{|\text{corr}_{o_1,o_2}|}{|c_1\text{-match}|}
\]

\[
\text{MatchRatio}_{o_2} = \frac{|\text{corr}_{o_1,o_2}|}{|c_2\text{-match}|}
\]

The combined match ratio is defined similar to f-score, as follows.

\[
\text{CombinedMatchRatio} = \frac{2\times|\text{corr}_{o_1,o_2}|}{|c_1\text{-match}|+|c_2\text{-match}|}
\]

Match ratios that are too high indicate concepts mapped to many other concepts, a suggestion of low precision. Match ratios close to 1.0 indicate the highest precision. Kirsten states that match ratios between 2-9 indicate reasonable levels of precision.

We use these set of metrics to estimate precision and recall in our large-scale alignment results. We also perform random spot checks of the resultant alignments to strengthen confidence in the results.

VII. OPTIMIZING ONTOLOGY ALIGNMENT WITH A BRANCH AND BOUND APPROACH

To improve scalability of alignment, we developed an algorithm based on a branch and bound approach. This algorithm considers each concept in ontology \(o_i\) and semantically compares it with concepts in ontology \(o_j\) seeking to eliminate subtrees in ontology \(o_j\) that are not likely to align. The algorithm moves depth first through the graph of \(o_j\); if a concept in \(o_j\) is not semantically close to the concept under consideration in \(o_i\), it is pruned from further consideration along with all of its children. Once the pairs of concepts to align are selected, alignment begins. This algorithm is designed to reduce the time complexity such that \(O(n^2)\) is a maximum; the actual complexity is typically much less, depending on the number of semantically close concepts across-ontology. This Branch and Bound approach is detailed in Figure 2.

The most important component of this algorithm is the method by which we evaluate whether two concepts are “semantically close”. Semantic closeness is determined by the relationship that we seek to acquire in the alignment. For example, if we seek to align using subclass relationships, then “semantic closeness” might be defined to be the existence of hyponymy relations among words in class names cross-ontology. If we seek to align using superclass relationships, then we might define “semantic closeness” to be the existence of hypernymy relations among words in class names cross-ontology. Our goal was to select a feature set that would prune unlikely candidates, resulting in a computationally inexpensive yet still highly accurate approach. Depending on the feature selected, we could expect to encounter varying execution times as well as varying accuracy results. As a starting point to align with subclass, we selected a feature that measures how many hyponyms occur between class names cross-ontology.

The evaluation was performed by aligning the Mammalian Phenotype (mp) Ontology and the Biological Process subset of the Gene Ontology (gobp) using subclass relations, with bag of words hyponymy used as the semantic distance check. As shown in Figure 3, the algorithm results in a dramatic reduction in execution time over the original alignment algorithm. The Branch and Bound algorithm aligns in an average of approximately 38 hours as compared to an average of 96 hours with the original alignment algorithm. On average, we found that time complexity was cut by approximately 1/3.

To complete our assessment of the value of the Branch and Bound algorithm, we evaluated whether the algorithm results in a reduction in accuracy. To do that, we compared the performance of the Branch and Bound algorithm with the performance of the original method when aligning the mp and gobp ontologies. To measure accuracy, we used the approximation metrics described in section VI. The evaluation was performed with the SVM models built for the Biomedical
FOR each root concept $c_j \in o_j$
    push $c_j$ onto stack;
ENDFOR
FOR each concept $c_i \in o_i$
    WHILE (not done)
        IF (stack is empty)
            done = true;
        ELSE
            $c_j = \text{pop stack}$;
            IF (semanticallyClose($c_i, c_j$))
                pairsToCompare.add($c_i, c_j$)
                push direct subclasses of $c_j$ onto stack;
            ENDIF
        ENDIF
    ENDWHILE
ENDFOR
Align(pairsToCompare)

Fig. 2. branchAndBoundAlign($o_i, o_j$)

Fig. 3. Branch & Bound and Alignment Performance Comparison for up to 1000 Pairs, including time to select pairs to align.

domain in an earlier part of this research. We averaged the performance over 3 SVM models. The semantic distance test was performed using the features that measures how many hyponyms appear between words in each class name. A comparison of the average match coverage metrics, which estimate recall, are shown in Figure 4. These results show that estimated recall is significantly lower, particularly for the second ontology. This is not entirely unexpected, given the nature of the Branch and Bound algorithm; that is, we might expect that less classes in ontology 2 are matched, since they are being pruned in the process. Recall from section VI that Kirsten recommends match coverage to be above 60%; however, we believe that in this case, low match coverage may not necessarily be an indicator of poor recall. However, we will still seek to improve the match coverage of the Branch and Bound algorithm. Match ratios, on the other hand, are satisfactory, as shown in Figure 5. This suggests that precision is not negatively impacted by the Branch and Bound approach. Recall from section VI that we seek match ratios in the range of 2-9, with 1 being considered a perfect score. Therefore, these results are quite encouraging. Given the nature of the Branch and Bound algorithm, we would expect that precision should remain as high as the original approach, since the alignment process remains constant. These estimated metrics were validated with random spot checks of the alignments by biomedical researchers. Their results were favorable with roughly 80% of the alignments being correct.

The key to improving recall of the Branch and Bound algorithm is to select the optimal feature for use in pruning; that is, the feature that is the best at determining “semantic closeness”. This feature or set of features should result in higher recall but not significantly increase processing time. Therefore we performed experiments to identify optimal semantic distance features for use in the Branch and Bound algorithm for subclass alignment. We compared the following features to determine if they would improve recall.

- Hyponyms between words in class names cross-ontology (used in original results above)
- Hyponyms between words in class names cross-ontology
- Synonyms between words in class names cross-ontology
- Hyponyms and hypernyms between words in class names cross-ontology

The results are shown in Figure 6. Again, these experiments involved aligning the mp and gobp ontologies. We found that
using the feature to detect hyponyms between class terms worked the best of all the single features. However, this was the feature originally used, therefore recall was not improved. Combining hypernyms and hyponyms worked best overall, delivering a slight improvement in recall. However, even with the combination of features, recall remains quite low. The use of match coverage as an estimation of recall may be flawed however, since the measure assumes all classes must be matched. With this limitation in mind, we believe that utilizing domain resources is the likely solution to optimizing recall for the Branch and Bound algorithm. The disadvantage, of course, is that the solution will not work in all domains.

In the case of the Stoilos String metric, the Branch and Bound algorithm execution time was significantly higher, more than 50% higher, therefore, we abandoned that approach for semantic closeness. In addition, we didn’t expect that a string metric would yield good results for semantic distance. In all other cases, execution time was not significant increased over the results discussed earlier in this section; in fact, in all cases, execution time remained within 5% of the original optimization.

VIII. EXPERIMENTAL RESULTS: ALIGNMENT OF LARGE SCALE BIOMEDICAL ONTOLOGIES

In this section, we present the results for aligning large-scale ontologies from the Open Biomedical Ontologies collection using subclass relations. We aligned 8 pairs of ontologies using the SVM models developed in earlier work. Recall that we use match coverage and match ratios to estimate recall and precision, respectively. A comparison of match coverage can be seen in Figure 7. In most cases, the alignment performs well. In particular, the NCI Thesaurus (nci) and Mouse Anatomy (ma) alignments result in strong match coverage, as do alignments of the Mammalian Phenotype (mp) and Mouse Anatomy (ma). This is especially encouraging since the SVM models were trained on mp,gobp examples, primarily. Match coverage performance on mp,gobp is also very good.

Match coverage for pairs mp, gomf and mp, gocc is not as strong. An analysis of the results revealed that in fact, many features were extracted for class pairs. For example, in the mp,gomf pair, features were extracted in well over 8000 class pairs. However, in most cases, only single features were identified. The SVM was trained with positive examples that typically possessed multiple features. Therefore, this suggests that the SVM should be trained with a more diverse set of data. In addition, further analysis is needed to identify features that will result in better match coverage for a broader set of ontologies in the Biomedical domain. However, it should also be noted that Kirsten’s metric is a very rough estimate, since it assumes that all classes much have a match. Clearly, that is not a reasonable expectation in a complex domain such as Biomedical.

Match ratio comparisons of large-scale alignment using subclass relations are shown in Figure 8. Again, in most cases, estimated precision looks very good. This is especially promising given that the SVM models were trained on mp,gobp examples, primarily. The estimated precision in the pair mp, gomf is troublesome; the results show that an inordinate number of matches occur in mp, an unexpected result. An analysis of the results showed that in fact, too many single classes were being matched to multiple classes, an indicator of low precision. Random spot checks confirmed low precision. This suggests that additional features should be identified and incorporated into the SVM to improve precision in a broader set of class pairs. However, overall, estimated precision is within the acceptable range (2-9), per Kirsten. In fact, Kirsten’s precision results were in the range of 8-46 in the Biomedical domain [11]. This suggests that our approach is highly precise.

Note that we supplemented these results with spot checks of the alignments by biomedical researchers. Data to spot check was selected randomly over the large sets of pairs. The results were favorable in general, with roughly 80% of the alignments being successful. This is an encouraging result and strengthens our confidence in the use of the estimation metrics.

IX. CONCLUSIONS

Solutions to scaling ontology alignment are only starting to emerge and in 2009, scaling alignment remains a challenge. In fact, scalability continues to be one of the grand challenges of ontology construction and use [8] and scaling ontology alignment in particular remains a research area [7].
We analyzed some approaches in this paper; in particular, we looked closely at Hu’s approach to ontology partitioning using a divide and conquer approach [9]. Hu’s algorithm does not consider the qualitative nature of links between classes and the execution time of partitioning is not reported. While pushing the complexity of alignment to an a priori process is a good approach (as it takes steps toward real-time alignment), we found that Hu’s approach is too expensive computationally.

We briefly described our approach to ontology alignment using non-equivalence relations. To scale ontology alignment, we presented an algorithm that is based on a branch and bound approach. In this novel approach, the algorithm considers each concept in the first ontology and compares it, depth first, to concepts in the second ontology. If the concept pairs are not “semantically close”, then the concept in ontology \( y \) is pruned from further consideration as are its children. This approach caps the maximum time complexity at \( O(n^2) \), and on average, we found that time complexity was cut by \( 1/3 \).

Since reference alignments for large-scale ontology alignments are not available and not feasible to create, we used emerging metrics from [11] to approximate precision and recall. We found that the Branch and Bound approach did not significantly reduce precision but it did reduce recall. We worked to identify functions to optimize the tradeoff between execution time and recall. We found that hyponymy relations between class names worked best as a single feature to identify semantic closeness. Identification of hyponymy relations in conjunction with hypernymy relations cross-ontology worked best to improve recall, without significant increase in execution time. Precision remained at fairly high levels in these experiments.

Finally, we present large-scale alignments of biomedical ontologies using subclass relations. In most cases, the algorithms performed well, according to estimated metrics and random spot checks. Alignment performed well even on ontologies that were not used in training the SVMs, such as the nci,ma and ma,nci pairs. In a few cases, we did find that performance was not satisfactory, as in the case of the mp,gomf and mp,gocc pairs. Analysis of the results showed that performance could be improved in two ways: (1) by adding new features to the SVM (2) by training the SVM with a broader set of training examples. In general, however, our results suggest that our approach is feasible for use in real-world applications, in terms of time complexity and accuracy.

**References**


