From Mappings to Modules: Using Mappings to Identify Domain-Specific Modules in Large Ontologies

Amir Ghazvinian
Stanford University
Stanford, CA, USA
amirg@stanford.edu

Natalya F. Noy
Stanford University
Stanford, CA, USA
noy@stanford.edu

Mark A. Musen
Stanford University
Stanford, CA, USA
musen@stanford.edu

ABSTRACT

The problem of ontology modularization is an active area of research in the Semantic Web community. With the emergence and wider use of very large ontologies, in particular in fields such as biomedicine, more and more application developers need to extract meaningful modules of these ontologies to use in their applications. Researchers have also noted that many ontology-maintenance tasks would be simplified if we could extract modules from ontologies. These tasks include ontology matching: If we can separate ontologies into modules based on the topics that these modules cover, we can simplify and improve ontology matching. In this paper, we study a complementary problem: Can we use existing mappings between ontologies to facilitate modularization? We present a novel approach to modularization based on mappings between ontologies. We validate and analyze our approach by applying our methods to identify modules for National Cancer Institutes Thesaurus (NCI Thesaurus) and Systematized Nomenclature of Medicine–Clinical Terms (SNOMED-CT).

Categories and Subject Descriptors

I.2.4 [**Artificial Intelligence**]: Knowledge Representation Formalisms and Methods—*Relation systems*

Keywords

ontologies, knowledge bases, ontology mapping, modularization

1. INTRODUCTION

The field of biomedicine has embraced the Semantic Web probably more than any other field. Ontologies in biomedicine facilitate information integration, data exchange, search and query of heterogeneous biomedical data, and other critical knowledge-intensive tasks [14]. As a result, there are many biomedical ontologies covering overlapping areas of the field [2]. Many of these ontologies are large: For instance, the National Cancer Institute's Thesaurus (NCI Thesaurus) [15] has 80,000 classes and Systematized Nomenclature of Medicine–Clinical Terms (SNOMED-CT), one of the key biomedical ontologies, has almost 400,000 classes [16].

Creating mappings among ontologies by identifying similar classes is a critical step in integrating data and applications that use different ontologies. With these mappings, for example, we can link resources annotated with terms in one ontology to resources annotated with related terms in another ontology, discovering new relations among the resources themselves (e.g., linking drugs and diseases). Because identifying mappings among ontologies manually is an enormous task, development of algorithms that try to find candidate mappings automatically is a very active area of research [4]. In recent years, Semantic Web researchers have developed many sophisticated algorithms that use a wide variety of methods, such as graph analysis, machine learning, and use of domain-specific and other background knowledge. For example, Hu and Qu [7] use a partitioning method to identify corresponding blocks of entities in two ontologies.

In our previous work [6] we have shown that large ontologies, such as those found in the field of biomedicine, pose a serious scalability challenge to advanced mapping algorithms. Many of these algorithms would be able to perform well if we were first able to identify modules of the larger ontologies in which the mappings are likely to lie.

In this paper, we examine the complementary problem. Given a set of mappings between a small, specific ontology and a large, broader ontology, we want to identify a module within the larger ontology that corresponds to the content of the small ontology. For example, a developer wants to use one of the large standard ontologies in biomedicine, such as NCI Thesaurus. Using such ontologies may be a requirement from users or funders, or the developer might want to use terms from such an ontology to facilitate inter-operability with other applications. However, the developer may need only a small portion of the 80K classes in NCI Thesaurus. For instance, a developer working on an application that analyzes data related to the cell cycle, may need only the part that is relevant to describing the cell cycle. Using a smaller module lets the developer use an ontology of a manageable size in her application while preserving the references to and the structure of the original ontology. Researchers have developed a number of algorithms for identifying logical modules within ontologies [17]. These methods often make use of the structure or properties of the ontologies to partition the ontology into modules (e.g., work by Cuenca Grau and colleagues [3]). Additionally, some of these methods use input from users to obtain modules [11]. In this paper, we present a technique for creating modules by using mappings between a large ontology and a domain-specific ontology that covers the domain that the user is interested in (e.g., cell cycle).

We use mappings from a number of biomedical ontologies to NCI Thesaurus and SNOMED-CT to identify modules within these large ontologies. We hypothesize that a large number of smaller biomedical ontologies map to subsections of the larger NCI Thesaurus and SNOMED-CT ontologies, thereby extracting domain-specific modules of these two ontologies. This paper makes the following contributions:

- We implement a method for ontology modularization based on mappings between ontologies.
- We validate and analyze this method by applying it to find modules for NCI Thesaurus and SNOMED-CT.

2. MATERIALS AND METHODS

We will now describe how we use mappings to identify ontology modules. Our methods consist of three main components: 1) generating mappings from a source ontology to the target ontology that we wish to modularize, 2) clustering mappings within the target ontology, and 3) using mapping clusters to identify modules within the target ontology.

We start with two ontologies:

1. the ontology that we wish to modularize as *modularization target*, or *target*, for short; and
2. an ontology covering some part of the domain that the target ontology covers, which we call the *source ontology*.

For example, NCI Thesaurus may be the target ontology, and an ontology covering the cell cycle domain, such as the Cell Cycle Ontology (CCO) [1] is the source ontology. We can use the mappings between CCO and NCI Thesaurus to identify a module within NCI Thesaurus that is relevant to the cell cycle. We report here on the more interesting variations of our modularization algorithm based on qualitative evaluation of the modules they produce.

2.1 Generating Mappings

To identify ontology modules in our target ontology, we first need a set of mappings between the source and the target ontology. Our method is independent of the method used to create the mappings, although we have not yet studied the effect of particular mapping methods on the outcome of the modularization. For our experiments, we use a simple lexical method to create the mappings. Our earlier research [5, 6] showed that this method is very effective for biomedical ontologies, and produces mappings with high precision and good recall. This method also scales easily to very large ontologies. Specifically, we create a mapping between two classes from different ontologies if the preferred name of one class matches the preferred name or synonym of the other class after normalization. Figure 1A shows two ontologies and the set of mappings between them. We refer to classes in the target ontology that are part of the mapping set as *mapping targets*.

2.2 Clustering

Once we have mappings to a target ontology, we cluster the mapping targets to identify critical regions of the target ontology for our module (Figure 1B). For scalability reasons, we clustered the mapping targets only in cases where our mapping algorithm found fewer than 4000 mappings from the source ontology to the target ontology.

In order to perform the clustering of classes within an ontology, we need a distance metric that can be applied to return a distance between any two classes in the ontology. We chose an edge-based semantic similarity metric developed by Pekar and colleagues [13]. Pekar et al's semantic similarity metric is defined as follows:

$$sim(c_1, c_2) = \frac{\delta(c_a, root)}{\delta(c_a, root) + \delta(c_1, c_a) + \delta(c_2, c_a)}$$

where c_a is the lowest common ancestor of class c_1 and c_2 and $\delta(c_1, c_2)$ is the length in number of edges of the longest distance between c_1 and c_2. We chose this metric for two reasons: it computes similarity based on distance within the ontology and it counts sibling terms lower down as closer together than siblings near the root. Because we actually require a distance metric and not a similarity metric in order to cluster classes, we take the inverse of the similarity metric described above.

We use the k-means clustering algorithm [10] to cluster the mapping targets within the ontology using the inverse similarity metric to compute distance between any two classes. The number of clusters is an input parameter to the algorithm.

2.3 Using Clusters to Identify Ontology Modules

For the pair of source and target ontologies (such as NCI Thesaurus and CCO), the clustering algorithm returns a series of clusters. We use these clusters to identify components of the target ontology that would constitute the module. We define a *component* as part of a subtree in the target ontology that is covered by one or more clusters. A *module* consists of one or more disconnected components. For example, when identifying a module of NCI Thesaurus relevant to the cell cycle using the Cell Cycle Ontology, our algorithm produced components anchored at the classes "Gene Product",

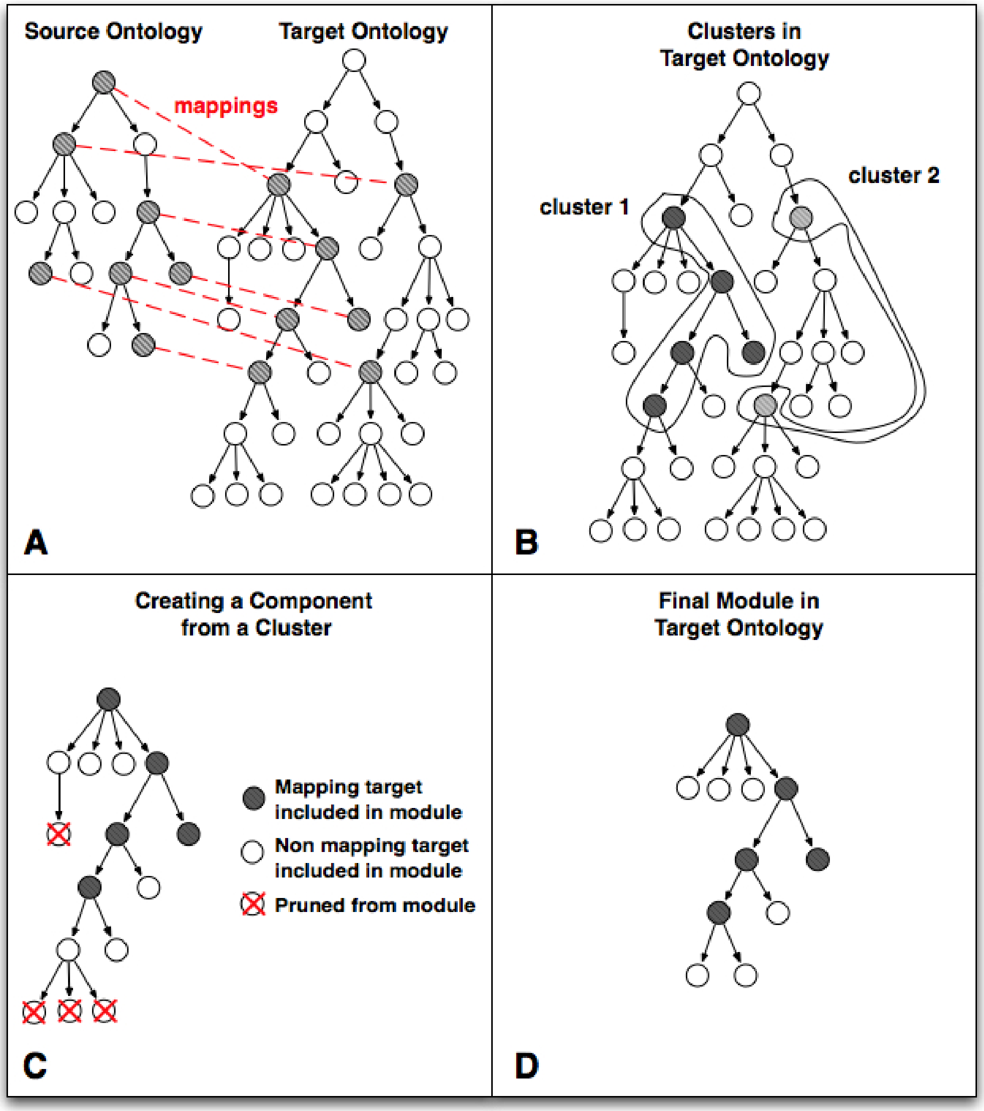
Source Ontology
Target Ontology
mappings
A
Clusters in Target Ontology
cluster 1
cluster 2
B
Creating a Component from a Cluster
Mapping target included in module
Non mapping target included in module
Pruned from module
C
Final Module in Target Ontology
D

Figure 1: The process of identifying modules using mappings between ontologies. Figure A shows the mappings between a source ontology and a modularization target. Figure B shows two clusters returned by the clustering algorithm. One cluster is light gray in color while the other is dark gray. When determining a module based on these clusters, we discard the light gray cluster since the mapping targets within that cluster are too sparse. Figure C illustrates the process of pruning the ontology subtree for the remaining cluster, which we use to create the module. We begin at each leaf and traverse the tree toward the root, removing all classes that are not mapping targets or direct children of mapping targets. Once we reach such a class, we stop pruning along that branch. Figure D displays the final module.

“Biological Process”, and “Anatomical Structure, System, or Substance.” The union of these components gives the module of NCI Thesaurus for the cell cycle domain.

Figure 1C illustrates the process of identifying components and then modules from clusters. For each cluster, we first determine the lowest common ancestor of all classes in the cluster. This common ancestor becomes a *candidate anchoring point* for a component. In order to determine whether or not this class does in fact serve as a good anchoring point for the component, we compute what percent of classes within the subtree below the anchor class are also mapping targets. If we find enough mapping targets in the subtree, we include in our final module a component anchored at that class. The number of mapping targets that we consider to be enough here depends on the percentage of the subtree covered by the mapping targets and ranges from 1 for subtrees with 100% coverage to 30 for subtrees with less than 10% coverage. Intuitively, this process allows us to remove the subtrees with just a few not closely related mapping targets inside a subtree: these subtrees are unlikely to become meaningful module components.

After we determine all anchors for the module, we prune each subtree to get a more specific and well-defined module. In many cases, all the mapping targets are clustered near the root of the subtree, because the representation of the domain in the source ontology does not go to the same level of detail as the representation in the target ontology. In order to get a cluster that corresponds more closely to the domain of the source ontology, we prune the lower levels of the component subtree that have no mapping targets in them. Specifically, we begin at each leaf in the components subtree and traverse the subtree toward the anchor, removing all the classes that are neither mapping targets themselves nor direct children of mapping targets. Once we reach such a class, we stop the pruning along that branch. Figure 1D illustrates the final module after pruning.

3. VALIDATION AND ANALYSIS

We applied the methods above to identify modules for NCI Thesaurus and SNOMED-CT. These ontologies are very commonly used in the biomedical community. Indeed, these two ontologies are currently listed as the two most viewed ontologies in BioPortal, a repository of more than 200 biomedical ontologies [12].[1] These two ontologies are also very large: NCI Thesaurus has almost 80K classes and SNOMED-CT has more than 380K. As source ontologies for modularization, we used 141 ontologies in BioPortal, with 106 of these ontologies containing fewer than 5,000 classes. Our process extracted 71 modules for NCI Thesaurus and 68 modules for SNOMED-CT. We were not able to extract modules in some cases for a variety of reasons: because the mapping set did not contain enough mappings to find relevant modules, the mapping set contained too many mappings to cluster, or the mappings were distributed too sparsely within the target ontology.

We use the modularization of these two ontologies in order to validate our algorithm and to analyze the process itself and the resulting modules in a number of ways: First, we examine some sample modules and their representative terms in order to understand the types of modules that our algorithm creates and to determine whether or not these modules are likely to be useful in an application setting (Section 3.1). Second, we analyze the ways in which the modules vary depending on the number of clusters given as input to our clustering algorithm (Section 3.2). Finally, we analyze characteristics of the modules, such as how well our mapping targets cover the module (Section 3.3).

3.1 Sample Modules

Figure 2 shows sets of representative terms for two of the modules generated by our algorithm. We generated both modules using 10 clusters as the input to our clustering algorithm. In the first example (Figure 2A), we identified a module of NCI Thesaurus that is relevant to electrocardiograms (EKG) using the Electrocardiography Ontology in BioPortal. The module consists of 61 classes, representative samples of which are shown in the figure. Of the 61 classes in this module, 41 (67%) are mapping targets. Additionally, the module makes up 100% of the subtree "EKG Concept" in NCI Thesaurus. In other words, the pruning process did not remove any classes. Figure 2B shows a SNOMED-CT module that includes the part of SNOMED-CT covering the same domain as the Mouse Adult Gross Anatomy. The module consists of 6,461 classes, of which 1,577 (24%) are mapping targets. This module had a significant amount of pruning: the module makes up only 25% of the subtree "Anatomical Structure" in SNOMED-CT.

In case of the anatomy module, even though both the subtree percentage and the mapping target coverage are lower than for the EKG module, we still seem to have identified a useful module for dealing with mouse anatomy. Because we prune, we also greatly decrease the size of the ontology, making it much more manageable and removing the very specific classes that are not relevant given the scope of the source ontology used to create the module.

Our experience shows that some combination of the absolute number of mapping targets as well as the coverage of the module are good indicators of how meaningful and useful the module is. We explore these characteristics further in Section 3.3.

3.2 From Clusters to Modules

Several of the modules that result from our algorithm have more than one component. For example, we already mentioned in Section 2.3 that when identifying a module of NCI Thesaurus relevant to the cell cycle using the Cell Cycle Ontology, we extracted the module that contains components anchored at the classes "Gene Product," "Biological Process," and "Anatomical Structure, System, or Substance."

We now analyze how particular characteristics of the modules and the components that constitute the module vary as a function of the number of clusters used as an input parameter for our clustering algorithm. Specifically, we look at the size, number of distinct components, and depth of components. Figure 3 shows how these characteristics change as the number of clusters changes, and also shows the number of modules we were able to identify from the 141 BioPortal ontologies we used as source ontologies.

First, as the number of clusters given as input to the k-means clustering increases, so, too, does the number of ontologies for which we are able to provide modules (Figure 3A). We see this trend because, as the number of clusters increases, the specificity of each cluster increases. When trying to identify which clusters are relevant for components of the final module, we discard very sparse clusters. Some ontologies have fairly specific components but this feature can be masked if there are not enough clusters to differentiate the component from other mapped targets in the ontology. So as we get more specific clusters, more components get included and therefore more ontologies.

[1] `http://bioportal.bioontology.org`

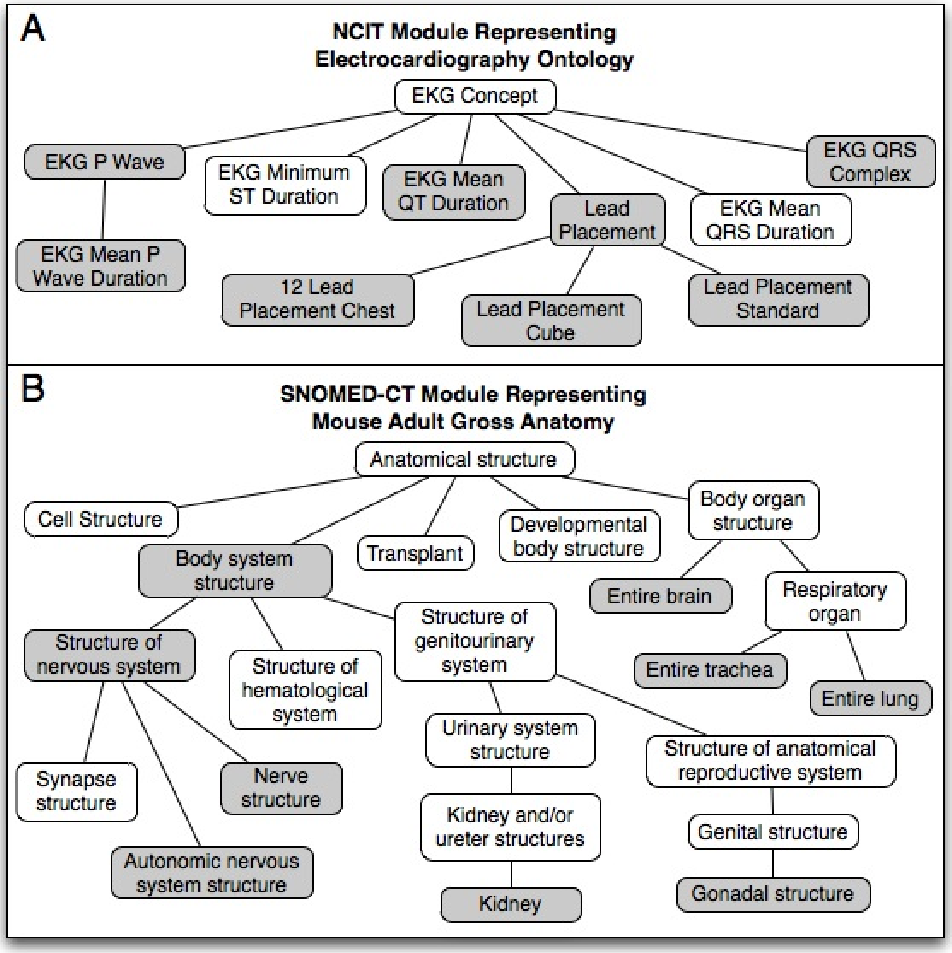

Figure 2: Sample modules and some of their representative terms. In each figure, the classes in gray boxes represent mapping targets and classes in white represent classes that were not mapping targets, but are included in the module through our algorithm. Note that the figure shows only parts of the modules. Figure A shows a module we identified within NCI Thesaurus that represents the domain of the Electrocardiography Ontology. Figure B shows a module discovered in SNOMED-CTthat is relevant to Mouse Adult Gross Anatomy.

Second, we observe in Figure 3B that the average size of modules increases as the number of clusters increases. This observations corresponds to the fact that the average number of module components goes up as the number of clusters increases. Though components get smaller and more specific, the modules consist of more components on average (Figure 3C), so module size increases.

Third, we observe a tradeoff in the number of clusters and module shapes and sizes. As number of clusters increases, so, too, do the number of components (Figure 3C) and the average depth of our modules (Figure 3D). Therefore, we find increasingly specific modules, but lose some of the scope that ties these disconnected components together.

For example, at both 10 and 20 clusters for NCIT, we have a module corresponding to Common Terminology Criteria for Adverse Events (CTCAE). However, at 10 clusters, the module consists of a single component with the root class “Adverse Event”, while at 20 clusters we find a module with several components representing the actual types of adverse events such as “Adverse Event Associated with Nervous System” or “Adverse Event Associated with Cardiac Arrhythmia.” Though we find more specific components that reflect the actual types of adverse events found in CTCAE, we lose the big picture, namely that all of these things are adverse events. Pruning wisely can help us avoid dealing with this problem as much because we can prune out the adverse event subtrees that are not relevant to CTCAE.

We see the same type of result in the anatomy modules, where we get a single module of “Anatomical Structure” when there are few enough clusters that all anatomical classes get clustered together, but we may get distinct components related to brain or heart anatomy if those parts of the anatomy are over-represented in the ontology. It is difficult to know which type of module is useful to the user and it probably varies depending on a specific application. For instance, for

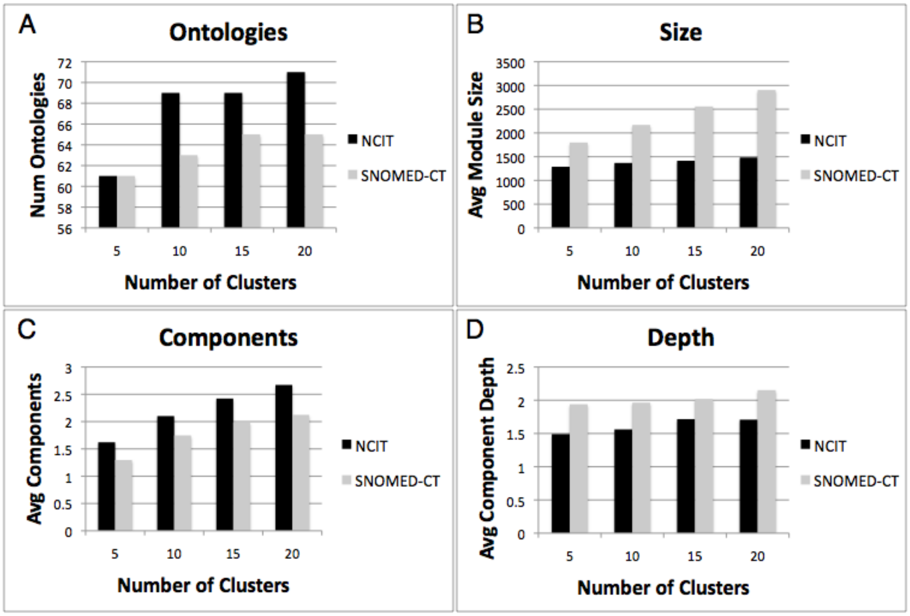

Figure 3: Characteristics of modules vary with the number of clusters used to generate the modules. A: the number of ontologies for which we are able to identify modules; B: the average size of the modules we identify; C: the average number of components per module; D: the average depth of the anchor class of each module component.

a general anatomy ontology, a module user would probably want the module rooted at "Anatomical Structure", whereas for a more specific ontology like BIRNLex, which deals with brain anatomy, you might want the module to contain only the anatomy relevant to the brain and would then want it to consist of just the "Brain" component.

Finally, these statistics provide a way to examine differences in the structure of SNOMED-CT and NCIT. NCIT has more components, but SNOMED-CT has a greater average depth per component. This difference indicates that there are many higher level classes for SNOMED-CT that are not very specific in scope and therefore are not as good for making modules as those classes which are at a greater depth. SNOMED-CT has much larger modules than NCI Thesaurus on average, which is not surprising because SNOMED-CT is a much larger ontology and therefore has more classes in the domains of these modules. For example, SNOMED-CT has more than 25,000 classes in the "Anatomic Structure" subtree, while the corresponding "Anatomic System, Substance, or Structure" subtree in NCI Thesaurus has about 6,000 classes. Thus we see that anatomy-related modules in SNOMED-CT are much larger than in NCI Thesaurus: for Mouse Adult Gross Anatomy, we identified a SNOMED-CT module with 6,461 classes, but the NCIT module we identified for the same ontology only contains 1,795 classes.

3.3 Module Coverage

In this section, we analyze the proportion of the module that is composed of mapping targets and the proportion of the ontology subtree that the module consumes.

Figure 4 shows a distribution of coverages for our modules. On average, our mapping targets cover 18% of the module for NCI Thesaurus and 16% of the modules for SNOMED-CT. Therefore, on average, more than 4/5 of the classes in the modules that we extracted are not included in the mapping set used to generate that module. The lowest mapping target coverage of any module identified by our algorithm is 3.4% for NCI Thesaurus and 2.9% for SNOMED-CT.

We also examine, for different values of percentage p, the number of modules that have mapping targets for at least p% of their classes (Figure 4A). Though we identified a comparable number of modules for both NCI Thesaurus and SNOMED-CT, the modules of NCI Thesaurus tend to be composed of a greater percentage of mapping targets than those of SNOMED-CT. For example, 30 of the NCI Thesaurus modules we identified consist of at least 30% mapping targets while only 14 SNOMED-CT modules boast the same mapping target coverage.

Next, we analyzed the percent of classes in the overall ontology subtree that are also contained in our module. This statistic measures how much of the subtree has been pruned to generate the module. On average, modules in NCI Thesaurus cover 30% of the overall subtree while modules in SNOMED-CT cover 27%. Two modules for SNOMED-CT and seven modules for NCI Thesaurus include the entire subtrees (i.e. no pruning). The minimum proportion of a subtree taken up by any module is 0.6% for both NCI Thesaurus and SNOMED-CT. Our algorithm pruned these modules heavily; Therefore, they do not take up a large part of their respective trees.

In Figure 4B, we show the number of modules that contain

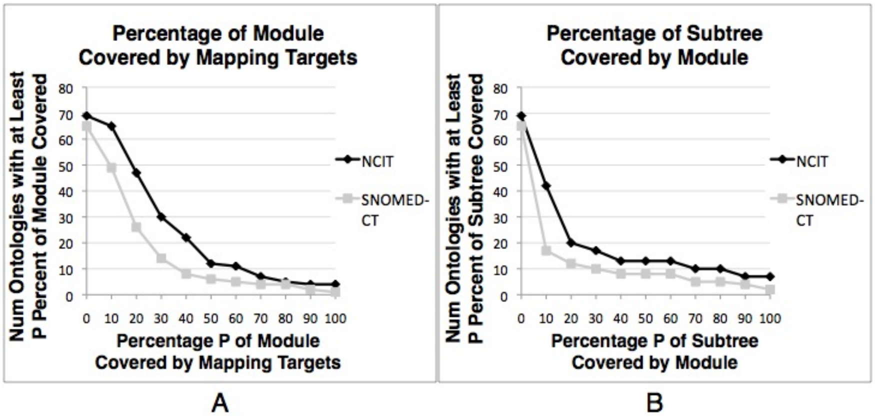

Figure 4: The coverage of modules. Figure A shows the percentage of our modules that are covered by mapping targets. The x-axis represents a percentage P and the y-axis shows how many of our modules have mapping targets for at least P% of their classes. Figure B shows the percentage of the overall ontology subtree is contained within our module. Intuitively, it is a measure of how much has been pruned from the module. In Figure B, the x-axis represents a percentage P and the y-axis shows how many of our modules cover at least P% of their respective subtrees.

at least $p\%$ of the original subtree, for different values of p. We see that the vast majority of modules for SNOMED-CT contain less than 10% of the original subtree and in fact only 17 of the 65 modules we identified for SNOMED-CT contain more than 10% of the subtree. By contrast, NCI Thesaurus has a much slower drop-off rate, with 13 modules that have contain at least 60% of the original subtree.

The coverage statistics for our modules could serve as a proxy for usefulness of the module. The percent of the module made up of mapping targets reveals how well our mapping set (and thus our original ontology) reflects the classes selected for the module. Analyzing the percent of the ontology subtree covered by our module demonstrates both whether the module comprises a substantial portion of the relevant ontology subtree and how well we can prune out the portions of the subtree that are not relevant for the module. Though we do not formally evaluate the modules, these statistics show that our algorithm can produce modules with both good coverage of the relevant ontology subtree and that consist of significant shares of mapping targets.

4. DISCUSSION

We have presented an approach that uses mappings between ontologies in order to extract domain-specific modules from large ontologies based on their mappings to smaller ontologies. In our experiments with NCI Thesaurus and SNOMED-CT, using the ontologies from BioPortal as the sources for mappings, we have identified a number of useful modules. We found that one of the key hurdles that we must overcome, is to find a way to determine how good a particular module is. Indeed, the same problem is true for most modularization approaches [17]: many authors discuss computational properties of their modules, but do not evaluate how useful these modules are to users. In our case, the requirements for extraction are driven by domain coverage of the module rather by its computational or structural properties. Thus, the problem of evaluating whether the module satisfies the user requirements is similar to the problem of ontology evaluation in general: how do we know that an ontology is useful for a specific class of applications [18]? Evaluation of an ontology depends on the specific applications for which it is being used; the same can be said for modules. Therefore, we plan to submit the modules that we have identified to BioPortal to enable the user community to use the modules in their applications, review them and comment on them. We hope that this crowd-sourcing approach to evaluation will enable us to get a better sense of what modules the users find useful for their applications. However, our initial evidence, which we present in this paper, indicates that our approach can indeed find interesting domain-specific modules.

We use the ontologies and the mappings to them essentially as *background knowledge*, similar in spirit to the way SAMBO [9] and ASMOV [8] use background knowledge (UMLS Metathesaurus, WordNet) to improve the quality of mappings themselves.

Providing modules as we have done in this paper can improve the performance of many applications. For applications with large ontologies, it can make the ontology more manageable and usable by including only relevant portions of the ontology. Providing modules relevant to multiple smaller ontologies can facilitate data integration for applications requiring the domains of those multiple ontologies. Instead of using multiple ontologies and having mappings between them for integration, we can simply use the union of multiple modules within the same ontology, which are essentially automatically integrated.

Although we present here a way to identify modules of large ontologies that correspond to the domains of smaller ontologies, our approach is not limited to cases where one ontology is large and the other ontology is small. We can use the same method to identify shared domains among large ontologies.

Additionally, in this paper, we utilized only subsumption relationships to identify modules, because those relationships were present in the ontologies we analyzed, but a similar approach could be used to traverse other types relationships.

Our approach has certain limitations. Because we use a simple lexical matching method, our results are limited to the domain of biomedicine and other domains, such as Cultural Heritage [19], where such a mapping method works well. In other domains, where class definitions do not contain rich lexical information, one will need to find scalable tools that would produce enough mappings to enable statistically significant analysis. Furthermore, our previous analysis [6] has shown that recall for a lexical matching algorithm in the field of biomedicine is about 65%. Scalable ways to increase recall, such as mapping composition, would give more mappings to input for the clustering algorithm, which would in turn likely give a better picture of the modules.

Additionally, because we are dependent on mappings to identify modules, in cases where there are not many mappings to a target ontology, we cannot identify relevant modules. For NCI Thesaurus, we were unable to identify modules in 50% of cases. For SNOMED-CT, we were unable to identify modules in 54% of cases. In the opposite scenario, when we have too many mappings, clustering these mappings becomes infeasible. In this paper, we were only able to cluster mappings in cases where we had 4,000 mappings or fewer between ontologies.

Furthermore, some of the modules we identify have a very low percentage of mapping targets, meaning many of the classes included in these modules were not mapping targets. For example, 16 of the modules we identified for SNOMED-CT consist of less than 10% mapping targets, Further research into better ways to prune the module might improve our coverage and our modules.

5. CONCLUSIONS AND FUTURE WORK

We have presented an approach that uses mappings between ontologies for modularization. Our analysis shows that this approach extracts modules that are specific to a particular domain. Varying the parameters of the algorithm controls the number of components in the modules and the specificity of the module itself. In the future, we plan to analyze the effect of the quality of the mappings and the number of mappings on the results of the modularizations. We will use the BioPortal user community to evaluate the usefulness of the modules that our approach produces and their use in applications.

6. ACKNOWLEDGMENTS

This work was supported in part by the National Center for Biomedical Ontology, under roadmap-initiative grant U54 HG004028 from the National Institutes of Health

7. REFERENCES

[1] E. Antezana, E. Tsiporkova, V. Mironov, and M. Kuiper. A cell-cycle knowledge integration framework. In *Data Integration in the Life Sciences*, v. 4075, pp. 19–34. Springer, 2006.

[2] O. Bodenreider and R. Stevens. Bio-ontologies: current trends and future directions. *Brief Bioinform*, 7:256–274, 2006.

[3] B. Cuenca Grau, I. Horrocks, Y. Kazakov, and U. Sattler. Modular ontologies. chapter Extracting Modules from Ontologies: A Logic-Based Approach, pages 159–186. Springer-Verlag, Berlin, Heidelberg, 2009.

[4] J. Euzenat and P. Shvaiko. *Ontology matching*. Springer, Berlin ; New York, 2007.

[5] A. Ghazvinian, N. F. Noy, C. Jonquet, N. Shah, and M. A. Musen. What four million mappings can tell you about two hundred ontologies. In *ISWC '09: Proceedings of the 8th International Semantic Web Conference*, pages 229–242. Springer-Verlag, 2009.

[6] A. Ghazvinian, N. F. Noy, and M. A. Musen. Creating mappings for ontologies in biomedicine: Simple methods work. In *AMIA Annual Symposium*, San Francisco, CA, 2009.

[7] W. Hu and Y .Qu. Block matching for ontologies. In *ISWC '06: Proceedings of the 5th International Semantic Web Conference*, pages 300–313. Springer-Verlag, 2006.

[8] Y. R. Jean-Mary, E. P. Shironoshita, and M. R. Kabuka. ASMOV: Results for OAEI 2009. In *Workshop on Ontology Matching at ISWC 2009*, 2008.

[9] P. Lambrix, H. Tan, and Q. Liu. Sambo and sambodtf results for the ontology alignment evaluation initiative 2008. In *Workshop on Ontology Matching at ISWC 2008*, 2008.

[10] J. B. MacQueen Some Methods for classification and Analysis of Multivariate Observations. In *Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability*, Berkeley, University of California Press, 1:281-297.

[11] N. F. Noy and M. A. Musen. Modular ontologies. chapter Traversing Ontologies to Extract Views, pages 245–260. Springer-Verlag, Berlin, Heidelberg, 2009.

[12] N. F. Noy, et.al. BioPortal: ontologies and integrated data resources at the click of a mouse. *Nucleic Acids Research*, 37(Web-Server-Issue):170–173, 2009.

[13] V. Pekar and S. Staab. Taxonomy learning: factoring the structure of a taxonomy into a semantic classification decision. In *19th Int. Conf. on Computational Linguistics*, pages 1–7, Morristown, NJ, USA, 2002. ACL.

[14] D. L. Rubin, N. H. Shah, and N. F. Noy. Biomedical ontologies: a functional perspective. *Brief Bioinform*, 9(1):75–90, January 2008.

[15] N. Sioutos, S. de Coronado, M. Haber, F. Hartel, W. Shaiu, and L. Wright. NCI Thesaurus: A semantic model integrating cancer-related clinical and molecular information. *Journal of Biomedical Informatics*, 40(1):30–43, 2007.

[16] K. Spackman, editor. *SMOMED RT: Systematized Nomenclature of Medicine, Reference Terminology*. College of American Pathologists, Northfield, IL, 2000.

[17] H. Stuckenschmidt, C. Parent, and S. Spaccapietra, editors. *Modular Ontologies: Concepts, Theories and Techniques for Knowledge Modularization*. Springer-Verlag, Berlin, 2009.

[18] Y. Sure, A. Gmez-Prez, W. Daelemans, M. Reinberger, N. Guarino, and N. Noy. Why evaluate ontology technologies? because it works! *IEEE Intelligent Systems*, 19(4):74–81, 2004.

[19] A. Tordai, J. van Ossenbruggen, and G. Schreiber. Combining vocabulary alignment techniques. In *K-CAP '09: 5th Int. Conf. on Knowledge Capture*, pages 25–32, 2009. ACM.