Combining Ontology Modules for Scientific Text Annotation

Marcus Albert Alves da Silva[1,2], Maria Claudia Cavalcanti[1]

[1] Military Institute of Engineering (IME), Brazil
[2] Brazilian Army Technological Center(CTEx),Brazil
m_albert@ig.com.br; maryoko@gmail.com

Abstract. Semantic annotation enables the inclusion of semantic information (ontology concepts), readable through software agents, into the most varied types of documents available on the Web. This operation facilitates search and access to relevant information. However, it is not an easy task, since the content to be annotated can be associated to multiple large ontologies. This article presents an approach based on the association of techniques and methods of ontology modularization, that enables the construction of a structure composed of concepts over one or more ontologies. This reduced structure is useful for automated semantic annotation of scientific texts. In addition, the structure attends a research specific interest. In order to evaluate this approach, a software tool was implemented. An experiment in the biomedical field, on a Corpus of 500 scientific papers, was conducted and showed good results, confirming the applicability of this approach.

Categories and Subject Descriptors: H.2.4 [**Database Management**]: Textual Databases; I.2.4 [**Computing Methodologies**]: Knowledge Representation Formalisms and Methods

Keywords: information retrieval, ontology modularization, semantic annotation, semantic web

1. INTRODUCTION

Nowadays, the Semantic Web is the focus of specialists from different research fields. The reason for such interest is the possibility of processing the Web content with the help of computer resources and semantics. There is a prediction that the content of the Semantic Web will grow considerably in the next years [Berners-Lee et al. 2010]. In this scenario, ontologies have been used to map Web content to its meaning, i.e., a knowledge representation that makes explicit a non-ambiguous view of a group of researchers about a knowledge domain. In particular, the semantic annotation contributes to the Semantic Web as it inputs hidden semantic information (such as ontology concepts) into Web pages and other kinds of documents that are available on the Web.

Semantic annotation is also useful in the scientific scenario, where a lot of scientific discoveries are "hidden" within text content. These texts can be found in traditional databases [Xiao and Eltabakh 2014], associated to each tuple, and are usually small observations associated to the scientific finding that the tuple represents. On the other hand, scientific findings can also be found in formal scientific articles, available at digital libraries, such as PubMed [1]. These articles are much larger than database annotations. Moreover, since articles are typically multidomain, their annotation demands more than one ontology.

The Biomedical area has been investing heavily on ontologies [Smith et al. 2007; Whetzel et al. 2011]. Biomedical ontologies are known for their large size in terms of number of classes. There are some that have more than 500.000 classes. These rich structures provide a detailed view of a

[1] http://www.ncbi.nlm.nih.gov/pubmed/

This work has been partially supported by CNPq (307647/2012-9) and FAPERJ (E- 26/111.147/2011).

knowledge domain. On the other hand, the use of such ontologies turns the semantic annotation task complex and expensive, demanding high performance computers to make it viable. Also, it is worth noting that biomedical scientific texts include concepts that are related to distinct knowledge domains, which means that to reach a more efficient semantic annotation, it is necessary the use of multiple ontologies. These ontologies usually include concepts that point to other ontologies. Through these "points", also known as "mappings", it is possible to traverse, combine or reuse ontologies, as if they were a single ontology. However, dealing with more than one large ontology demands a great deal of computational resources, not easily available to the user. In this work, we focus on the automatic semantic annotation of scientific texts, which can be very time consuming, as reported by Belloze [2013].

There are many initiatives that apply ontology modularization to facilitate their reuse [Parent and Spaccapietra 2009; Ghazvinian et al. 2011; Simperl 2010]. Some of these initiatives focus on the semantic annotation task [D'Aquin et al. 2006; Wennerberg et al. 2011; Gomes 2012]. The latter ones take into account the identification of the user research interest. However, as far as we could investigate, none of them aim at the automatic annotation of a large set of a scientific articles. On the other hand, Xiao and Eltabakh [2014] deal with a large set of small texts on a scientific database, proposing a summarization method. Differently from our approach, they do not focus on formal scientific articles, nor address the difficulties in dealing with large and multiple ontologies.

The present work proposes a set of steps as a systematic way to build a unified structure that consists of a set of modules of multiple-ontologies. It involves the use of ontology modularization techniques, as well as the identification of the user annotation interest. The main contribution of this approach is to turn viable (agile) the reuse of multiple ontologies for automatic semantic annotation of a Corpus of scientific texts, on regular computers.

This work is organized as follows. Section 2 presents some basic concepts on semantic annotation and ontology modularization. In section 3 some related work are briefly described and discussed. The proposed approach is presented in Section 4. An experiment, using the proposed approach, is described and its results are discussed in Section 5. Section 6 summarizes the contributions and points to future work.

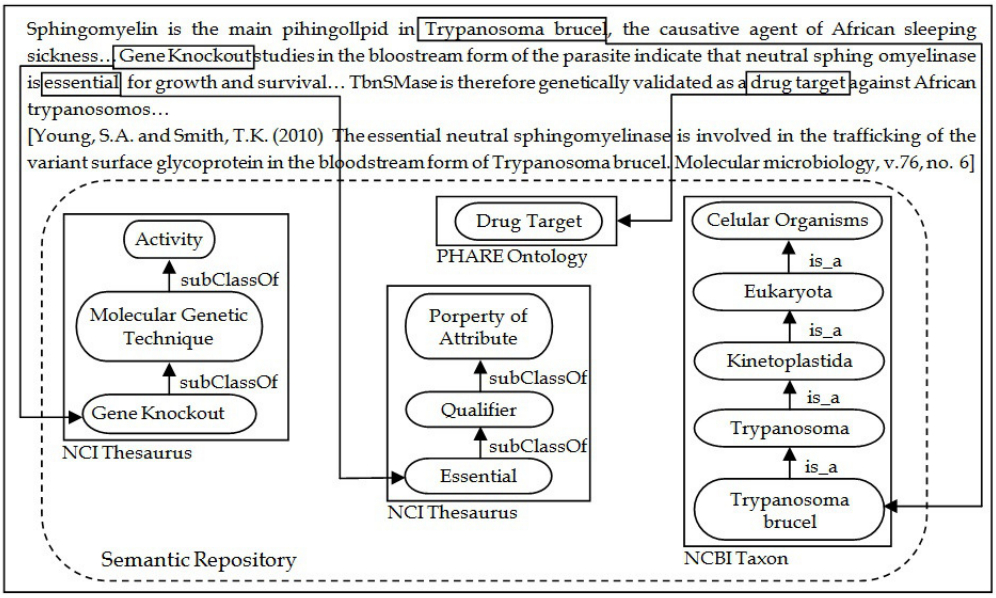
2. SEMANTIC ANNOTATION AND ONTOLOGY MODULARIZATION

The focus of this work is on the application of ontology modularization techniques to facilitate the semantic annotation task. Some basic but important concepts related to this work are presented in the following subsections.

2.1 Semantic Annotation

Semantic annotation allows the input of metadata or ontology concepts into texts. It is the association of relevant text expressions to concepts and/or instances of an ontology. An annotation should be well defined, not ambiguous and easy to understand by domain specialists, in a way that it could be useful for the information retrieval process [Gomes 2012]. Ontologies are typically built focused on a single domain. Therefore, for a scientific text to be well-annotated, the use of multiple ontologies or taxonomies is required. This is especially true for the texts of the biomedical area. An excerpt of one of these texts is shown in Figure 1. It illustrates possible annotations using three different ontologies. The expression ”Drug Target” was annotated with the PHARE ontology, the expression ”Tripanosoma brucei” was annotated with the NCBI Taxonomy, and finally, the expressions ”essential” and ”gene knockout” were annotated with the NCI Thesaurus.

The annotation process can be intrusive or not. It is intrusive when the annotation is inserted inside the document under annotation. It is non-intrusive when the annotation is registered externally (e.g.



Sphingomyelin is the main pihingollpid in Trypanosoma brucel, the causative agent of African sleeping sickness... Gene Knockout studies in the bloostream form of the parasite indicate that neutral sphing omyelinase is essential for growth and survival... TbnSMase is therefore genetically validated as a drug target against African trypanosomos...
[Young, S.A. and Smith, T.K. (2010) The essential neutral sphingomyelinase is involved in the trafficking of the variant surface glycoprotein in the bloodstream form of Trypanosoma brucel. Molecular microbiology, v.76, no. 6]

Fig. 1. Associating text expressions to ontology concepts

in a database). The annotation process may be manual (conducted by an specialist) or automated (carried out by a software tool). In both cases, when large ontologies are used, i.e., ontologies with more than 300 thousands concepts, such as some biomedical ontologies, the process becomes difficult and costly. In order to minimize this problem, some works suggest the use of a reduced version of these ontologies, i.e., a structure with a reduced number of concepts. However, to find out which concepts should compose such reduced structure is not an easy task. The next subsection presents the concept of ontology modularization.

2.2 Ontology Modularization

In areas such as Biomedicine, where there are numerous and large ontologies, the application of modularization techniques can be useful. However, in order to put modularization into practice greatly depends on the goals that are pursued, and thus, reducing ontologies into smaller modules has to follow some criteria [Parent and Spaccapietra 2009]. It is worth noting that ontologies are semantic-based structures, in which each class and each property have different meanings. Such differences should be taken into account in the process of modularization, i.e., certain classes and/or properties (relationships) may be more relevant than others, and should be preferred in the generated modules [Garcia et al. 2012].

There are two main approaches for ontology modularization: ontology partitioning and ontology module extraction. The partitioning divides the ontology in groups of concepts where members are semantincally close. This approach may be applied in situations where the ontology must be divided to facilitate its use and maintenance. On the other hand, the module extraction approach is used to reduce the ontology to a domain subset of interest, for a specific application [D'Aquin et al. 2006]. This approach, also known as segmentation, is useful to identify a subset of concepts that cover a specific sub-domain, representing an important and reusable part of the ontology [Noy and Musen 2009].

3. RELATED WORK

This work involves a combination of concepts of semantic annotation, ontology construction and modularization techniques. First we have investigated some related work that aim at building structures that represent part of ontologies, which involve ontology construction and modularization techniques. On the other hand, some other related work bring interesting ideas on how to use/create such modular ontologies for semantic annotation.

3.1 Building structures that represent part of ontologies and/or combine them

There are already many tools for extracting ontology modules, but all of them are restricted to a single ontology. PROMPT [Noy and Musen 2004], which was implemented as a plugin to the Protege, is used to manually crop an ontology module. KMI, which was developed by the Knowledge Media Institute, takes into account inference during the extraction process [D'Aquin et al. 2006]. NeOn, which is an open source tool, is a general purpose tool for ontology engineering tasks, and includes plugins for ontology modularization [Doran et al. 2007]. Finally, there is the SEGMENTATION tool [Seidenberg 2009] that implements the transversal extraction technique, which uses a graph representation of the ontology. It initiates the extraction at a specific user-defined node (concept or set of concepts), and based on its relationships it builds a list of concepts for the extraction, preserving the semantic relationships between concepts during the extraction [Seidenberg 2009].

The approach proposed by Souza Jr. et al. [2010] aims at the construction of a structure named Emerging Ontologies (EO), which involves elements of more than one ontology. The idea is to provide a global view of several ontologies in one single structure, based on the mappings between them, and assuming the mapped concepts are the most familiar to the ontologies' users. The resulting structure depends on the number of mappings between the ontologies in hands, and may not represent a significative portion of none of the ontologies involved. Another work [Ghazvinian et al. 2011] also use the mappings between pairs of ontologies, aiming at the extraction of a module from the larger ontology of the pair. The main idea is to facilitate their maintenance, since biomedical ontologies can have more than 500,000 concepts. However, the resulting structure may not be useful for other purposes, than for local maintenance.

Another work that deserves attention [Queiroz-Sousa et al. 2013] aims at the summarization of an ontology. Although it is restricted to a single ontology, it also aims to facilitate the understanding of an ontology, by reducing it to a subontology composed of relevant and connected concepts. This work proposes a method based on measures of centrality in graphs to produce a summarized version of an ontology. But the most interesting aspect of this work is the idea of starting the module extraction (summary), based on a set of user-defined concepts.

A global view of multiple ontologies can be useful for semantic annotation with concepts that come from more than one ontology. Connecting ontologies' modules, i.e. through common concepts, can facilitate inferences in automated semantic annotation. However, none of the discussed approaches focus on semantic annotation. To build a relevant structure for semantic annotation, a previous analysis of text contents, combined with the identification of the user interest should be taken into account.

3.2 Using modular structures of ontologies for semantic annotation

A few related works were found that combined module extraction with semantic annotation. D'Aquin et al. [2006] propose a way to dynamically select, reduce and combine ontologies to annotate a current Web page. The Magpie [Domingue and Dzbor 2004] tool was adapted in order to provide this functionality. The idea is to select and annotate relevant concepts from the current Web page. However, the user is relieved from manually choosing a suitable ontology every time he wishes to browse new

Web content. Based on the selected relevant terms, the algorithm applies the traversal approach [Noy and Musen 2009], and relies on inferences to extract the ontology module. The extracted modules (from distinct ontologies) compose the final structure that is used to annotate the current Web page. The authors reported promising results through an experiment using one Web page and a large medical ontology. However, they do not address automatic annotation of a large set of scientific texts. Moreover, the authors do not report on performance when dealing with multiple large ontologies,.

A similar approach [Gomes 2012] assists the user at the manual annotation of scientific texts. It also relieves the user from the selection and module extraction work. The idea is to observe the user while annotating, to capture his/her interest while using a single ontology. After a few annotations, small fragments of the text surrounding the annotation are analyzed and relevant terms are selected. Based on the set of terms considered more relevant and the terms already annotated in the text fragment, modules from multiple ontologies are extracted. The Segmentation tool [Seidenberg 2009] is used for module extraction. Then, the modules are offered (recommended) to the user during the annotation activity. Although this work addresses scientific text annotation, its focus is on manual annotation and, therefore it does not address automatic Corpus annotation. Moreover, it uses multiple ontologies' modules, but does not combine them.

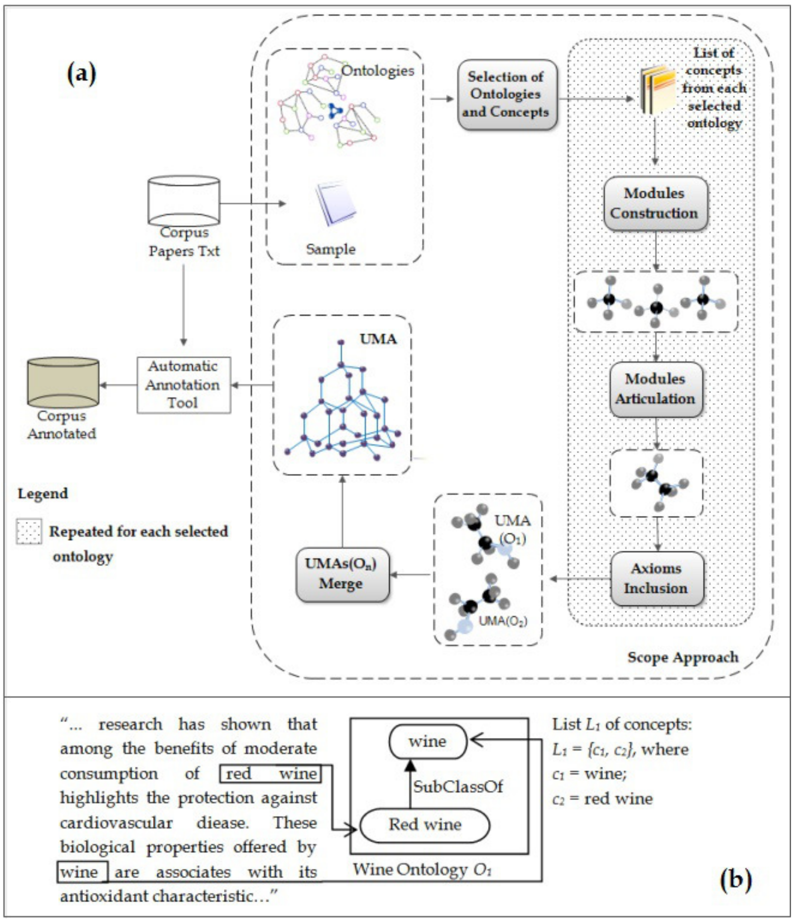
Another interesting work proposes the input of semantic content into medical image descriptions and patient reports is the challenge presented by Wennerberg et al. [2011]. Their approach aims to identify fragments of an anatomy's ontology based on relevant concepts to annotate medical imaging of patients suffering from lymphoma. The idea is to reduce the ontology through the application of well-defined rules, and linguistic and statistical techniques. However the authors report that relevant concepts initially indicated by statistical analysis, are too generic (close to the ontology root) and may generate large modules. Moreover, this approach also does not aim to address automatic scientific texts annotations.

4. AUTOMATIC SCIENTIFIC TEXT ANNOTATION APPROACH

In the context of scientific environments, where there are large ontologies and large sets of texts, this work presents a new approach for automatic text annotation (Corpus annotation). It consists of a set of steps that turn viable (agile) the reuse of multiple ontologies, without the need for high performance computer architectures. The main idea is to build a unified structure that results from the combination of a set of modules extracted from multiple-ontologies. It involves the use of ontology modularization techniques, as well as the identification of the user annotation interest.

As stated before, automatic semantic annotation of texts may be very expensive in terms of computational resources. At the beginning of a scientific research, a simple keyword-based query on the query interface of a digital scientific papers library can return more than a thousand hits. Moreover, when annotating such texts with large ontologies, a lot of not relevant annotation may occur. Therefore, in order to avoid such waste, the identification of the user annotation interest is required. In this sense, a user-defined set of ontology concepts may represent his/her research interest. However, when distinct large ontologies are involved, the identification of such interest may become a hard task. In the BioPortal Recommender [Jonquet and Musen 2010] environment, the user can input a small piece of text into the system, and it returns a recommendation about what ontologies and concepts may be used to annotate that text. However, a small piece of text is not representative of a Corpus. Therefore, in order to simulate the user research interest, the proposed approach take a user-defined subset (sample) of the whole set of selected papers (Corpus), and similarly to the BioPortal approach, it automatically annotates the sample.

Inspired by the existing modularization techniques, more specifically, based on the traversal approach, this work uses its own module extraction approach, aiming at producing articulated modules extracted from one ontology. The idea is to generate a reduced structure that can cover the user



"... research has shown that among the benefits of moderate consumption of red wine highlights the protection against cardiovascular diease. These biological properties offered by wine are associates with its antioxidant characteristic..."

List L_1 of concepts:
$L_1 = \{c_1, c_2\}$, where
c_1 = wine;
c_2 = red wine

Fig. 2. Approach Overview and step 1 in detail

research interest, and whose nodes (concepts) are connected to each other, keeping their semantic relationships.

4.1 Approach Overview

The proposed approach can be described in five main steps, as shown in Figure 2 (a). The first step is the **Selection of Ontologies and Concepts** that represent the user interest. Then, the following three steps are concerned with the generation of a reduced structure for each ontology: **Modules Construction(2)**; **Modules Articulation(3)**; and **Axiom Inclusion(4)**. These three steps are repeated for each Ontology initially selected (step 1). The last step, **UMA's Merge(5)**, combines all modules extracted from the set of Ontologies into a single structure.Each of these modules is described in more details as follows:

(1) **Selection of Ontologies and Concepts**: This step is responsible for the identification of the user's interest in the Corpus. It is a user task to choose some representative articles from the Corpus, according to his/her research interest. Based on this sample of articles, it is defined a set of ontologies $\Omega = \{O_1, O_2, \dots O_n\}$ that are relevant and useful for the semantic annotation of those articles. Then, the sample is annotated using an automatic semantic annotation tool with all the selected ontologies (one at a time). For each ontology O_i, the concepts used in the

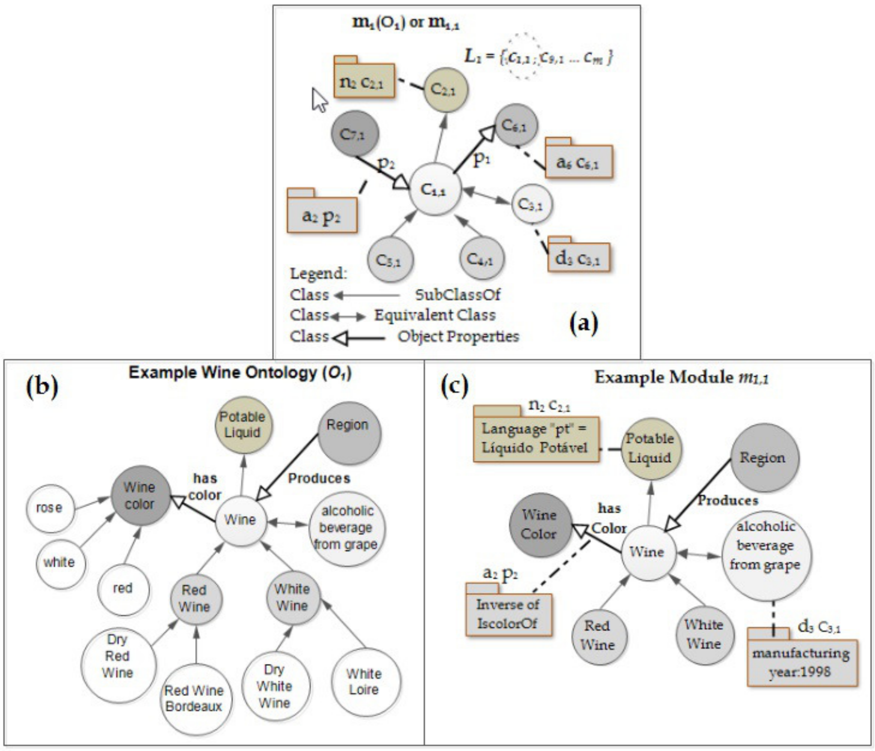


Fig. 3. A real world example of module construction

annotations are taken as the representation of the user interest and compose the list of concepts $L_i = \{c_1, \; c_2, \; \dots \; c_m\}$, where $0 < m \leq n$. In other words, L_i is the set of concepts from the ontology O_i that correspond to the user interest. A concept $c_j \in L_i$, which is extracted from O_i, is also referred to as $c_j(O_i)$, or simply $c_{j,i}$. Thus, at the end of this step, if more than one ontology is selected, there will be a list of lists $\Lambda = \{L_1, L_2, ...L_n\}$. Figure 2 (b) shows the annotation of a text fragment using the wine ontology (O_1). In this example, the terms "wine" and "red wine" were annotated with the concepts of the ontology O_1, and compose the list of selected concepts $L_1 = \{c_1, c_2\}$, where c_1 and c_2 correspond to "wine" and "red wine" concepts, respectively. The output of this step is the set of selected ontologies Ω and the set of corresponding lists of concepts Λ, which define the research interest and will be used in the next step. Each concept $c_{j,i} \in L_i$, for each $L_i \in \Lambda$, will be the core of the construction of a module in the next step of the approach. The following three steps are executed for each $L_i \in \Lambda$ and the corresponding $O_i \in \Omega$.

(2) **Modules Construction**: In this step, for each concept $c_{j,i} \in L_i$, a new module $m_{j,i}$ is generated. Besides the core concept ($c_{j,i}$) itself, $m_{j,i}$ is composed by other concepts $c_{x,i}$ from O_i that are connected to it. For each $c_{x,i}$, its corresponding data properties, annotations, and constraint axioms also come with it to constitute $m_{j,i}$. Formally, these elements are defined as follows.
Let $c_{j,i}$ be the core concept of a module $m_{j,i}$, which is defined according to $m_{j,i} := (C, P, D, N, A)$, where
—C is the set of concepts $c_{x,i}$ from O_i such that are hierarchically close (parents and children) to $c_{j,i}$, or that are connected to it through an object property $p_k \in P$;
—P is the set of object properties p_k from O_i such that p_k defines a relationship between two concepts $c_{x,i}$ and $c_{y,i}$, where $x = j$ or $y = j$, and $0 < k \leq |P|$;
—D is the set of data properties d_k from O_i such that d_k associates basic data types to a concept $c_{x,i} \in C$, where $0 < k \leq |D|$;

```
**Data**: A set of modules $M = \{m_1, m_2, ..., m_n\}$ from Ontology $O_1$
**Result**: The Structure named $UMA$ from $O_1$
**Begin**;
$UMA \leftarrow m_s$, where $m_i \in M$; **remove** $m_s$ from $M$;
**while** $M$ *is not empty* **do**
    **for *each*** $m_i \in M$ **do**
        **if** $m_i$ *is connected to* $UMA$ **then**
            $UMA \leftarrow m_i$; **remove** $m_i$ from $M$;
        **end**
    **end**
    **if** $M$ *is not empty* **then**
        **for *each*** $m_j \in M$ **do**
            **for *each*** $c_k \in C$ ***of*** $m_j$ **do**
                $m' \leftarrow$ createModule($c_k$);
                **if** $m'$ *is connected to* $UMA$ **then**
                    $UMA \leftarrow m'$; **exit For**;
                **end**
            **end**
            $UMA \leftarrow m_j$; **remove** $m_j$ from $M$;
        **end**
    **end**
**end**
**return** $UMA$;
**End**;
```

Algorithm 1: Articulation Algorithm

—N is the set of annotations n_k from O_i such that n_k decribes the meaning of a concept $c_{x,i} \in C$, where $0 < k \leq |N|$;

—A is the set of constraint axioms a_k such that $\exists p_k$ defined in a_k, $p_k \in P$ or $\exists c_{x,i}$ defined in a_k, $c_{x,i} \in C$, where $0 < k \leq |C|$.

Figure 3 (a), illustrates a module construction, where the module $m_{1,1}$ is built based on the core concept $c_{1,1} \in L_1$. This module is composed by a set of classes(concepts), properties, annotations and axioms from O_1. More specifically, it is included in $m_{1,1}$ concepts that are hierarchically close to it $(c_{2,1}, c_{3,1}, c_{4,1}, c_{5,1})$, the object properties p_1 and p_2 that connect $c_{1,1}$ to other concepts, and also their corresponding concepts $(c_{6,1}, c_{7,1})$. In addition, the module also includes: the data property d_3 of concept $c_{3,1}$; the annotation n_2, that describes the meaning or label of the concept $c_{2,1}$; and the restrictive axioms a_2 and a_6 that define rules over the object property p_2 and over concept $c_{6,1}$, respectively. A simple real world example using a partial view of the wine ontology is showed in figure 3(b), where the core module is the "wine" concept. the generated module is shown in figure 3(c). It is worth mentioning that this approach is based on the transversal extraction approach [Noy and Musen 2009], using deep level 1. Moreover, the generated modules are independent of each other, but may have concepts in common. For instance, a concept $c_{x,i}$ that is a subclass of the core concept of a module, may be related to the core concept of another module. In this case, $c_{x,i}$ is present in both modules.

(3) **Modules Articulation**: For each concept $c_{j,i} \in L_i$, a module $m_{j,i} \in M_i$ was generated in the previous step. Thus, $M_i = \{m_{1,i}, m_{2,i}, ...m_{m,i}\}$, where $0 < m \leq n$. This step aims to connect these modules through the identification of common concepts (points of articulation). It involves an investigation of the feasibility of including connections between concepts from different modules. If possible, it includes new concepts that provide this connection, i.e., that articulate two or more modules. At the end of this step, a new structure named UMA_i (Unified Modules for Annotation) for Ontology O_i is generated. As explained before, this step is repeated for each O_i. Therefore,

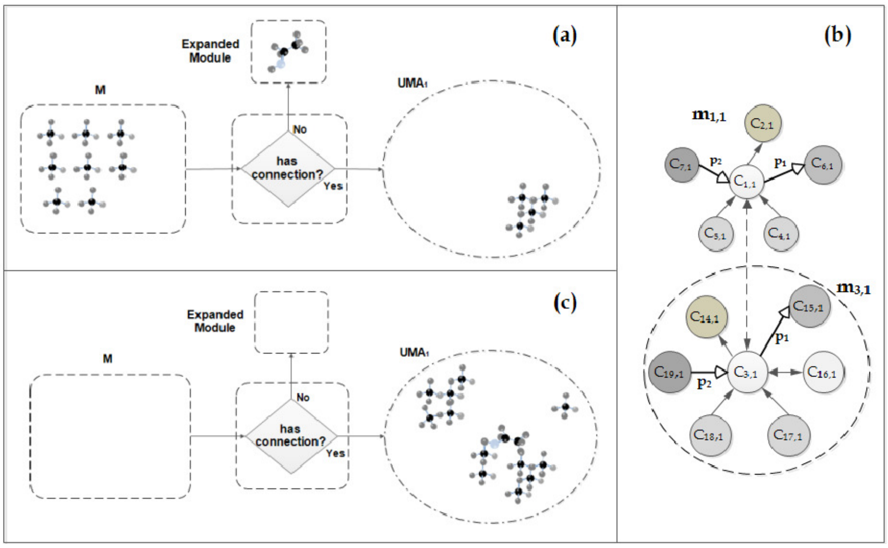


Fig. 4. Module Articulation

one UMA_i is created for each O_i. The UMA_i construction starts with an arbitrary module $m_{s,i}$.Algorithm 1 details how this step is processed. For each module $m_{j,i} \in M_i, j \neq s$, it checks to see if it is connected to (i.e., if it has concepts in common with) the UMA_i structure. if so, it is added to the UMA_i structure, and removed from M_i. As the UMA_i structure grows, the M_i list decreases. After verifying that M_i has only modules that do not have any connection to the UMA_i structure, then each of its remaining modules is expanded. The module expansion is illustrated in Figure 4(a). It consists of taking the set of concepts C of the module, as if it was a list L of concepts of interest, and execute the same procedure of the module construction step, generating a m'. In the example, concept $c_3 \in C$ of module $m_{1,1}$ is used to create a new model $m_{3,1}$. If, this new module connects to the UMA_i structure, both $m_{1,1}$ and $m_{3,1}$ are added to the UMA_1 structure, as shown in figure 4(b). If no m' is connected to the UMA structure, module $m_{1,1}$ will be added anyway. Note that while there are modules in M_i, then UMA_1 is a growing structure, and thus, there is a chance that it can be further connected to other modules inside the UMA_1 structure. At the end of this step, each module of the set M_i is now part of the UMA_i structure. It may be composed by three kinds of modules: connected modules, expanded connected modules and disconnected modules, as shown in Figure 4 (c). Redundancies (common concepts) are removed.

(4) **Axioms Inclusion**: In the Axiom Inclusion step the constraint axioms collected on the module construction step, which refer to some UMA_i concepts and/or properties, are added to it. Some constraint axioms may refer to concepts that are not in the UMA_i structure, i.e., concepts from O_i that were not included. Taking into account that these concepts are not part of the user interest, and that including them would grow UMA_i structure indefinetly, these axioms were discarded. Similarly to the module definition, the UMA_i structure can be formally defined according to: $UMA_i := (C, P, D, N, A)$, where
—C is the set of concepts $\{ c_{x,i} \mid \forall\ m_{j,i} \in M_i, \forall\ c_{x,i} \in C \text{ of } m_{j,i} \} \cup L_i$, i.e., all concepts from each and all modules $m_{j,i}$ union with the core concepts in L_i;
—P is the set of object properties p_k from O_i such that p_k defines a relationship between two concepts $c_{x,i} \in UMA_i$ and $c_{y,i} \in UMA_i$, where $0 < k \leq |P|$;
—D is the set of data properties d_k from O_i such that d_k associates basic data types to a concept $c_{x,i} \in C$, where $0 < k \leq |D|$;
—N is the set of annotations n_k from O_i such that n_k decribes the meaning of a concept $c_{x,i} \in C$,

where $0 < k \leq |N|$;

—A is the set of constraint axioms a_k such that $\forall p_k$ defined in a_k, $p_k \in P$ and $\forall c_{x,i}$ defined in a_k, $c_{x,i} \in C$, where $0 < k \leq |C|$.

(5) **UMA Merge**: As explained before, the three previous steps are executed for each $O_i \in \Omega$. Therefore, at the beggining of this step we have the set $\Upsilon = \{UMA_1, UMA_2, ...UMA_n\}$, where each UMA_i is a module extracted from O_i. In the UMA's Merge step, these UMA_i are merged, forming a final larger and unique UMA structure. The adopted merge strategy is the simplest one, i.e., UMA is defined as the tuple (C, P, D, N, A), where the C set is the union of the elements of all C sets from each $UMA_i \in \Upsilon$, and similarly, the same happens with the sets P, D, N, and A. It was not in the scope of this work dealing with the ontology alignment problem. This is the reason why UMA is not an ontology itself. It is just a unified structure that results from the composition of multiple and different ontology modules. It facilitates the automatic semantic annotation process of a Corpus, as it is an all in one single structure that unites concepts of different ontologies.

4.2 Prototype Implementation

A prototype named UMA Project (Unified Modules for Annotation Project) was implemented as a proof of concept of the proposed approach. The UMA Project tool was written in Java, version 1.7. It deals with the OWL format for ontology, since in the biomedical scenario, most of the ontologies are in RDF/OWL formats. Thus, for ontology file manipulation, it was used the OWLAPI version 3.4.5, and for reading RDFa annotations it was used JavaRDFa API. For the Merge step, the OWLAPI OWLOntologyMerger class was used, preserving each ontology concept identity (URI), and avoiding alignment. The implementation also included the use of existing tools, such as the annotation tool Autometa [Fontes et al. 2013] to perform automatic annotations, from which it obtains the list of selected concepts (step 1). An experiment using this implementation is reported in the next section.

5. EXPERIMENT AND DISCUSSION

This work was motivated by a specific scientific scenario of a Ph.D. thesis [Belloze 2013] that was developed at the Computational and Systems Biology Program from the IOC/FIOCRUZ, at Rio de Janeiro. The chosen research theme was on prioritizing drug targets, more specifically, the focus was on gene essentiality. The main idea was to extract information from scientific texts, which could report research on using techniques to find out gene essentiality for protozoa and model organisms. Although the semantic annotation showed useful results to identify new correlations, and thus facilitate the decision and prioritization of drug targets, it was very time consuming. This was especially true because the scientist dealt with a large set of texts and large ontologies. Although the exact times were not registered, annotations with the three whole ontologies, separately, on a set of more than 700 texts, took approximately 2 months to be completed.

In order to evaluate the proposed approach, an experiment was conducted using a set of texts and two ontologies from the same scientific scenario. The following relevant aspects were observed: (i) a possible reduction in computational cost during the process of automated semantic annotation and possible reduction for the execution time of this task; (ii) the possibility, even though using a reduced structure of the complete ontology, to achieve a good utilization compared to that annotation process with complete ontology for a selection of text indicated by the user.

The experiment described here was performed using a Dell Poweredge server with Intel Xeon E5-2420 1.90GHZs, 15MB Caches, 29Gb of RAM and 64-bit architecture. Using Linux Ubuntu 12.04.3 LTS 64-bit version operating systems, as well as the software tools described in the previous section. We selected five hundred (500) scientific articles from the PubMed portal (Portal maintained by the U.S. National Library of Medicine National Institutes of Health) whose content was related to biomedical

Table I. Annotation Times with Complete Ontologies

Experiment using Complete Ontologies		
Ontology	**Annotation Time for 10 papers**	**Estimated time for 500 papers**
NCBI Taxon	16hours 48min 27sec	35 days 22min 30 sec
NCI Thesaurus	10hours 47min 18sec	22 days 11hours 15min
Total time	**1 day 3 hours 35min 45 sec**	**57days 11hours 37min 30 sec**

Table II. Construction and Annotation Times from UMAs

Experiment using UMAs			
Structure	**UMA's Construction Time**	**Annotation Time for 10 papers**	**Estimated time for 500 papers**
UMA1	4hours 08min and 35sec	3hours 57min 38sec	8 days 6hours 1min 40sec
UMA2	7hours 52min and 41sec	8hours 10min 21 sec	17days 37min 30sec
UMA3	6hours 07min and 37sec	7hours 17min 38sec	15days 4hours 41min 40sec
UMA4	3hours 40min and 29 sec	3hours 35min 15 sec	7 days 11hours 22min 30 sec
UMA5	5hours 22min and 27sec	4hours 18min 32sec	8 days 23hours 26min 40 sec
Average	**5hours 26min 22 sec**	**5hours 27min 53sec**	**11 days 9hours 22min 30sec**

ontologies. The selected items were converted to text format (TXT) and composed the Corpus of the experiment. In this experiment, it was used the NCI Thesaurus ontology, version 11.06d (National Cancer Institute Thesaurus), that describes types of abnormal human cells, which may occur both in disease states as in disease models linked to cancer, and the NCBI Taxon ontology, version 1.2 - release 2009 (National Center for Biotechnology Information Taxon), ontology based in taxonomy of living organisms and associated artifacts. These ontologies have 89,131 and 392,448 concepts, and 223.6 Mb and 255.1 Mb approximate sizes of OWL files, respectively.

5.1 Comparing structures

From the Corpus, five different ramdom samples, composed of 10 papers, were generated. For each sample, two distinct lists of user interest concepts, L_1 and L_2, were built, based on the sample annotation with the selected ontologies, NCI Thesaurus (O_1) and NCBI Taxon (O_2), respectivelly. These lists were then used to generate modules UMA_1 and UMA_2, performing steps 2, 3 and 4 for each list. Step 5 generates the final UMA structure, which is the combination of structures UMA_1 and UMA_2. It is worth to note that, for all samples, the UMA structure showed a considerable reduction in the amount of concepts in relation to the total of the concepts in the two original ontologies. The UMA structures generated were on average 93% less than the full two ontologies combined.

5.2 Evaluating performance and time in automated semantic annotation task

The performance dealt with in this section, is associated with the processing time in performing the automatic annotation. Initially, it was measured the time of annotation on a sample of 10 papers in TXT format using the Autometa tool for automatic semantic annotation, with the full NCI Thesaurus ontology. Then the procedure was repeated on the same sample with the full NCBI Taxon ontology, where the results obtained were presented in Table I. Based on these numbers, it also presents an estimation of the time that would take to annotate the whole set of papers (500), which gets close to 2 months.This is coherent with the time taken by the scientist in the motivating scenario.

In order to evaluate the processing times using the UMA strutures, the annotation was performed using each of the 5 samples (10 paper samples), and using the corresponding UMA structure built for each sample. Note that each UMA structure is composed by the union of concepts of modules UMA_1 and UMA_2 extracted from NCI Thesaurus and NCBI Taxon ontologies, respectively. Thus during the automatic annotation process, it was annotated with concepts from both ontologies, in a single operation. The processing times presented in Table II show a reduction of approximately 80%,

Table III. Real time annotation using UMAs

Annotation Time 500 papers		
Structure	**05 Simultaneous Processes**	**Sequential**
UMA1	1day 18hours 05min 18sec	7days 18hours 39min 15 sec
UMA2	3days 10hours 26min 36sec	15days 8hours 11min 32sec
UMA3	3days 01hour 17min 13sec	13days 22hours 19min 37sec
UMA4	1day 13hours 28min 13sec	7days 01hour 18min 13sec
UMA5	1day 16hours 26min 32sec	7days 14hours 38min 01sec
Average	**2days 19min 27 sec**	**9days 02hours 13min 16sec**

Table IV. Indicators for usefulness analysis

Indicators	**Descriptions**
C_{int}	set of concepts of interest defined by the sample
C_{anot}	set of distinct concepts annotated throughout corpus
C_{ac}	set of distinct annotated concepts that belong to C_{int} where: $C_{ac} = \{C_{anot} \cap \ C_{int}\}$
C_{surr}	set of distinct annotated concepts that don't belong to C_{int} , but are surrounding the C_{int}, where: $C_{surr} = \{\ C_{anot}(\text{UMA}) - C_{int}\ \}$, and $C_{anot}(\text{UMA})$ is the C_{anot} using UMA
C_{out}	set of distinct annotated concepts that don't belong to C_{surr} and don't belong to C_{int}, where: $\forall c((c \in C_{out}) \leftrightarrow ((c \notin C_{surr}) \wedge (c \notin C_{int})))$, where c is an annotated concept $\in C_{anot}$
QC_{int}	quantity of concepts in the set C_{int}, where: $C_{int} = \mid C_{int} \mid$
Q_{ac}	quantity of concepts in the set C_{ac}, where: $C_{ac} = \mid C_{ac} \mid$
Q_{out}	quantity of concepts in the set C_{out}, where: $C_{out} = \mid C_{out} \mid$
Q_{surr}	quantity of concepts in the set C_{surr}, where: $C_{surr} = \mid C_{surr} \mid$
T_{anot}	quantity of concepts in the set C_{anot}, where: $T_{anot} = \mid C_{anot} \mid$
T_{xmaxA}	maximum rate of hit for structure, where: $T_{anot} = Q_{ac} \div QC_{int}$
T_{con}	total of concepts that compose the structure (Complete Ontology Or UMA).
uE	usefulness rate of the structure, where: $uE = (\ QC_{int} + Q_{surr}\ ) \div T_{con}$

if compaired to the processing times showed in Table I, for the annotation of both sets of 10 and 500 texts. It is worth to note that, even though in Table II as estimation is provided for the set of 500 texts, the real processing times showed in Table III are very close to the estimated numbers.

Despite the significant reduction of time, one must consider the time spent during the process of building the UMA structures. Processing times spent on building each UMA structure, seen in Table II, showed that on average, it took about 5-6 hours more, which does not really impact in the case of the annotation of a set of 500 texts.

During the annotation with the UMA structures, it was observed that both the server processor and the memory were not being fully required. Given this scenario, it was performed a new experiment, where the set of 500 articles in TXT format, were divided into five distinct groups containing 100 articles each. An application written in ShellScript language was built to enable the simultaneous execution of the five processes on the operating system, running the annotation of each group of 100 articles (leveraging the capabilities of the server). It can be observed in Table III, the significant reduction of about 80% in the duration of the whole process, if compared to the sequential annotation process.

5.3 Evaluating the UMA usefulness on the annotation process

Other aspects that worth noticing in the results obtained during the annotation task are the annotation usefulness (uE) and coverage ($Tx_{max}A$), with respect to the user interest, i.e., how much of the selected user interest concepts (QC_{int}), obtained on step 1, were used in the annotation, and how much of the UMA concepts ($Q_{ac} + Q_{surr}$) were used in the Corpus annotation. These indicators (metrics) are described in Table IV. Table V shows the results of such metrics, compairing the whole Corpus

Table V. Comparing Outcomes from the Corpus Annotation (Full Ontologies vs. UMA)

Interests (Samples)		**NCI Thesaurus Ontology**						**Concepts and Structures**	
Sample Id	QC_{int}	Q_{ac}	Q_{surr}	T_{anot}	T_{xmaxA}	Q_{out}	**uE**	T_{con}	**Structure**
1	2251	2196	-	7468	97.6%	2523	5.55%	89131	Complete Ontology
		2160	2749	4909	96.0%	-	23%	21647	UMA 1
3	2713	2590	-	7400	95.5%	2160	5.88%	89131	Complete Ontology
		2532	2650	5182	93.3%	-	22%	23272	UMA 3
4	2468	2414	-	7469	97.8%	2448	2.71%	89131	Complete Ontology
		2377	2607	4984	96.3%	-	23%	21478	UMA 4
NCBI Taxon Ontology									
Sample Id	QC_{int}	Q_{ac}	Q_{surr}	T_{anot}	T_{xmaxA}	Q_{out}	**uE**	T_{con}	**Structure**
1	74	67	-	1037	90.5%	869	0.04%	392448	Complete Ontology
		67	101	168	90.5%	-	3%	5833	UMA 1
3	106	95	-	1033	89.6%	807	0.06%	392448	Complete Ontology
		95	131	226	89.6%	-	1%	18326	UMA 3
4	78	64	-	1030	82.1%	873	0.02%	392448	Complete Ontology
		63	93	156	80.8%	-	3%	4789	UMA 4

annotation, for each sample (samples 1, 3, and 4), when using the corresponding UMA_i module and when using the full Ontology O_i.

With respect to the annotation usefulness (uE) with the UMA modules, note that the NCI Thesaurus ontology modules had more than 20% of usefulness while the NCBI Taxon had at most 3%. However, when compaired to the uE of the full ontologies, the corresponding modules (UMA) had a much better result in both cases, which means their usage avoided a significant waste of computer resources. With respect to the annotation coverage ($Tx_{max}A$), the rates achieved by all UMA modules in the annotation process are very close to what was achieved with the use of the corresponding full ontology. This means that the modules generated based on the Corpus samples, were good enough for the rest of the Corpus. This is a very good result, but it is probably also due to the way the Corpus was formed. The more homogeneous the Corpus is, the more efficient we get. Both usefulness and coverage depend on the ontology used in the annotation, i.e., on how close the ontology is to the Corpus. Analyzing table V it is clear that NCI Thesaurus is more closely related to the Corpus than the NCBI Taxon. This does not mean an ontology is better than the other, since they were chosen based on the annotation of a sample of articles. However, it explains the difference in the results found.

6. CONCLUSION

This work presents a new approach for automatic text annotation, in the scientific scenario, where the user face the challenge of dealing with a large set of texts and large ontologies. It combines the use of ontology modularization and merge techniques to facilitate and speed up the annotation process. In addition, it provides an agile way to identify the user annotation interest. The main idea is to build a unified and lighter structure that results from the combination of a set of modules of multiple-ontologies. An experiment with the annotation of a Corpus composed of 500 texts, showed good results. In terms of size, it achieved, on average, 93% less of the size of the full two ontologies combined. In terms of processing time, for the worst case, it achieved a reduction of about 30 days less than using the whole ontologies for the Corpus annotation. Finally, in terms of usefullness and coverage, the UMA structures were considerably more useful and provided a similar coverage, if compared with the annotation using the whole ontologies. Therefore, the main contribution of the proposed approach is its agility and efficiency on attending the user interest while annotating scientific Corpus. Future work includes improvements in the current approach to provide connected UMA structures, as much as possible. In addition, we plan to apply this approach on Corpus of different areas, with different ontologies, to demonstrate its wide applicability.

REFERENCES

Belloze, K. T. *Priorização de Alvos para Fármacos no Combate a Doenças Tropicais Negligenciadas Causadas por Protozoários (in Portuguese).* Ph.D. thesis, Fundação Oswaldo Cruz, Brazil, 2013.

Berners-Lee, T., Cailliau, R., Groff, J.-F., and Pollermann, B. World-Wide Web: the information universe. *Internet Research* 20 (4): 461–471, 2010.

D'Aquin, M., Sabou, M., and Motta, E. Modularization: a key for the dynamic selection of relevant knowledge components. In *Proceedings of the International Workshop on Modular Ontologies.* Athens, Georgia, USA, pp. 1–15, 2006.

Domingue, J. and Dzbor, M. Magpie: supporting browsing and navigation on the semantic web. In *Proceedings of the International Conference on Intelligent User Interfaces.* Funchal, Portugal, pp. 191–197, 2004.

Doran, P., Tamma, V. A. M., and Iannone, L. Ontology Module Extraction for Ontology Reuse: an ontology engineering perspective. In *Proceedings of the International Conference on Information and Knowledge Engineering.* Lisbon, Portugal, pp. 61–70, 2007.

Fontes, C. A., Cavalcanti, M. C., and de C. Moura, A. M. d. An Ontology-Based Reasoning Approach for Document Annotation. In *Proceedings of the IEEE International Conference on Semantic Computing.* Irvine, CA, USA, pp. 160–167, 2013.

Garcia, A. C., Tiveron, L., Justel, C., and Cavalcanti, M. C. Applying Graph Partitioning Techniques to Modularize Large Ontologies. In *ONTOBRAS-MOST-CEUR Workshop Proceedings.* Recife,Brazil, pp. 72–83, 2012.

Ghazvinian, A., Noy, N. F., and Musen, M. A. From Mappings to Modules: using mappings to identify domain-specific modules in large ontologies. In *Proceedings of the International Conference on Knowledge Capture.* Banff, Canada, pp. 33–40, 2011.

Gomes, P. C. C. *Suporte a Anotação Múltipla Baseada em Modularização de Ontologias: Uma Experiência na Área Biomédica (in Portuguese).* M.S. thesis, Instituto Militar de Engenharia (IME), Brazil, 2012.

Jonquet, C. and Musen, M. A. Building a Biomedical Ontology Recommender Web Service. *Journal of Biomedical Semantics* 1 (4): 1–18, 2010.

Noy, N. F. and Musen, M. A. Specifying Ontology Views by Traversal. In *Proceedings of the International Semantic Web Conference.* Hiroshima, Japan, pp. 713–725, 2004.

Noy, N. F. and Musen, M. A. Traversing Ontologies to Extract Views. In H. Stuckenschmidt, C. Parent, and S. Spaccapietra (Eds.), *Modular Ontologies.* Springer, Berlin, pp. 245–260, 2009.

Parent, C. and Spaccapietra, S. An Overview of Modularity. In C. P. Heiner Stuckenschmidt and S. Spaccapietra (Eds.), *Modular Ontologies.* Springer, Berlin, pp. 5–23, 2009.

Queiroz-Sousa, P. O., Salgado, A. C., and Pires, C. E. S. A Method for Building Personalized Ontology Summaries. *Journal of Information and Data Management* 4 (3): 236–250, 2013.

Seidenberg, J. Web Ontology Segmentation: extraction, transformation. In C. P. Heiner Stuckenschmidt and S. Spaccapietra (Eds.), *Modular Ontologies.* Springer, Berlin, pp. 211–243, 2009.

Simperl, E. Guidelines for Reusing Ontologies on the Semantic web. *International Journal of Semantic Computing* 40 (4): 239–283, 2010.

Smith, B., Ashburner, M., Rosse, C., Bard, C., Bug, W., Ceusters, W., Goldberg, L. J., Eilbeck, K., Ireland, A., Mungall, C. J., Consortium, T. O., Leontis, N., Rocca-Serra, P., Ruttenberg, A., Sansone, S.-A., Scheuermann, R. H., N, S., Whetzel, P. L., and Lewis, S. The OBO Foundry: coordinated evolution of ontologies to support biomedical data integration. *Nature Biotechnology* 25 (11): 1251–1255, 2007.

Souza Jr., H. C., de C Moura, A. M., and Cavalcanti, M. C. R. Integrating Ontologies Based on P2P Mappings. *IEEE Transactions on Systems, Cybernetics and Humans* 40 (5): 1071–1082, 2010.

Wennerberg, P., Schulz, K., and Buitelaar, P. Ontology modularization to improve semantic medical image annotation. *Journal of Biomedical Informatics* 44 (1): 155 – 162, 2011.

Whetzel, P. L., Noy, N. F., Shah, N. H., Alexander, P. R., Nyulas, C., Tudorache, T., and Musen, M. A. BioPortal: enhanced functionality via new Web services from the National Center for Biomedical Ontology to access and use ontologies in software applications. *Nucleic Acids Research* 39 (1): 541–545, 2011.

Xiao, D. and Eltabakh, M. Y. InsightNotes: summary-based annotation management in relational databases. In *Proceedings of the ACM SIGMOD International Conference on Management of Data.* Snowbird, USA, pp. 661–672, 2014.