Available online at www.sciencedirect.com

SCIENCE DIRECT®

Journal of Biomedical Informatics 39 (2006) 209–226

Journal of Biomedical Informatics

www.elsevier.com/locate/yjbin

# Semantic enrichment for medical ontologies

Yugyung Lee [a,*,1], James Geller [b,2]

[a] Department of Computer Science and Informatics, School of Computing and Engineering, University of Missouri, Kansas City, MO 64110, USA
[b] CS Department, New Jersey Institute of Technology, Newark, NJ 07102, USA

Received 10 June 2005
Available online 9 September 2005

**Abstract**

The Unified Medical Language System (UMLS) contains two separate but interconnected knowledge structures, the Semantic Network (upper level) and the Metathesaurus (lower level). In this paper, we have attempted to work out better how the use of such a two-level structure in the medical field has led to notable advances in terminologies and ontologies. However, most ontologies and terminologies do not have such a two-level structure. Therefore, we present a method, called semantic enrichment, which generates a two-level ontology from a given one-level terminology and an auxiliary two-level ontology. During semantic enrichment, concepts of the one-level terminology are assigned to semantic types, which are the building blocks of the upper level of the auxiliary two-level ontology. The result of this process is the desired new two-level ontology. We discuss semantic enrichment of two example terminologies and how we approach the implementation of semantic enrichment in the medical domain. This implementation performs a major part of the semantic enrichment process with the medical terminologies, with difficult cases left to a human expert.

*Keywords:* Terminology; Ontology; Semantics; Two-level ontology; Semantic enrichment; Semantic Web; Unified Medical Language System; Controlled Medical Vocabularies

## 1. Introduction

Terminological Knowledge Bases [1] are ontologies or terminologies that consist of an upper layer of semantic types (broad categories) and a lower layer of concepts. The two preeminent examples of terminologies with a two-level structure are the Unified Medical Language System (UMLS) [2–4] and the WordNet [5] mapping to the Suggested Upper Merged Ontology (SUMO) [6]. There are major differences between these two examples. The UMLS was built from the outset as a two-level structure, is about 16 years old and is used in the medical domain. On the other hand, WordNet is a general-purpose terminology which was developed independently from SUMO, and the mapping between these two knowledge structures was performed only recently. Even though the topic area of the UMLS is limited, it is an order of magnitude larger than the WordNet/SUMO combination. Because of this reason, and because the two-level structure of the UMLS was a design principle as opposed to being created after the fact, we will concentrate in our examples on the UMLS.

The Unified Medical Language System has been built over the past 16 years by the National Library of Medicine, with the help of a number of contractors. The UMLS must be considered a success by a number of criteria. One of these criteria is the ever-increasing size, which has reached over one million concepts. Another criterion is the widespread distribution of the UMLS in many organizations. The UMLS mailing list has about 600 members, some of them actively involved in online discussions. A third indicator of success is the large number of papers being published

[*] Corresponding author. Fax: +1 816 235 5159.
*E-mail address:* leeyu@umkc.edu (Y. Lee).

[1] This research was supported in part by the University of Missouri Research Board.

[2] This research was supported in part by the New Jersey Commission for Science and Technology.

1532-0464/$ - see front matter © 2005 Elsevier Inc. All rights reserved.
doi:10.1016/j.jbi.2005.08.001

about the UMLS. A Google Scholar search produces 2190 hits for the UMLS and 845 hits for the UMLS Metathesaurus.

While it is hard to say what exactly the reason for this success story is, we hypothesize that the two-level structure of the UMLS is an important contributing factor. The UMLS consists of three Knowledge Sources of which we are interested in two, the Metathesaurus [7,8] and the Semantic Network [9–11]. The *Metathesaurus* is a unified collection of many different medical terminologies. It is a compilation of terms, concepts, relationships, and associated information. The January 2004AC edition includes over 1 million concepts and 5 million concept names in over 100 biomedical source vocabularies.[3] The *Semantic Network* of the UMLS contains 135 semantic types (e.g., Disease or Syndrome, Virus). One may think of semantic types as high-level concepts, i.e., broad categories. These semantic types are organized in a hierarchy of IS–A links. The hierarchy consists of two trees, rooted in the semantic types Entity and Event. In addition, there are 53 kinds of non-IS–A relationships among these semantic types, e.g., *causes*, used in: Virus *causes* Disease or Syndrome. Every concept in the Metathesaurus is assigned to at least one, but often several, semantic types in the Semantic Network. One can say that a concept (in the Metathesaurus) is assigned some semantics by being assigned to a semantic type in the Semantic Network.

The design of the UMLS is far from intuitive. It raises a number of questions. For example, what is the precise distinction between semantic types and concepts? The examples in [12] show that the concept Diagnostic Procedure also occurs as semantic type Diagnostic Procedure. Thus, there cannot be a fundamental difference between concepts and semantic types. Another question is how to understand the exact nature of the assignment of concepts to semantic types. [13] offers the following explanation: "In fact, most Metathesaurus concepts are subtypes of their semantic type (e.g., "Salmonella" is a kind of Bacterium), while some are instances (e.g., "American Medical Association" is an instance of Professional Society")." However, both the Semantic Network and the Metathesaurus make use of IS–A links. Thus, if we allow for concept assignments to be identical to IS–A (subtype) links, the whole two-level structure is lost.

Most vexing is the following question (Fig. 1). Given a concept *c* that is assigned to a semantic type *X*, and a concept *d* that is an IS–A child of *c*, the UMLS structure requires that we assign *d* to a semantic type. Every concept must be assigned to a semantic type. However, if *c* is assigned to *X*, then, very likely, *d* will also be assigned to *X*. That makes the assignment of *d* to *X* redundant. Yet, the designers of the UMLS require such an assignment. This is even more paradoxical if one adds the following fact: If *X* IS–A *Y* in the Semantic Network, then the assignment of *c* to *Y* is prohibited by the designers of the UMLS, as concepts should be assigned to the most specific semantic type possible [14].

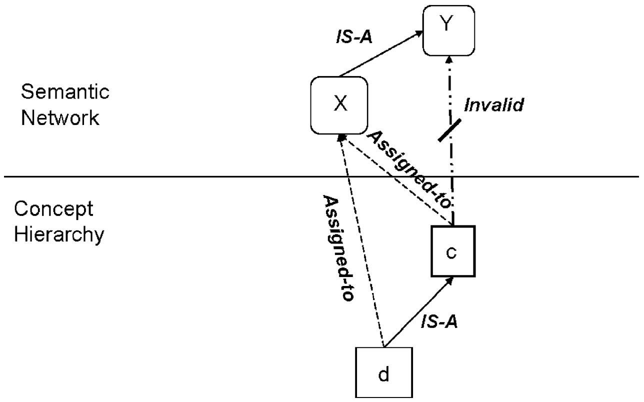

Fig. 1. UMLS two-level structure. The connection between *c* and *Y* is prohibited.

Our analysis of the UMLS and its two-level structure has led us to introduce a whole category of ontologies that imitate this two-level structure. We call these ontologies Terminological Knowledge Bases (TKBs) [1]. The work on mapping WordNet to SUMO [6] has strengthened our belief that such two-level structures are of universal interest. Their usefulness is not limited to the medical domain. We will discuss reasons why the two-level structure of the UMLS might have contributed to its success.

The second part of the paper is written under the assumption that the two-level structure is indeed advantageous. Most existing ontologies and terminologies are one-level structures. Creating upper-levels for these existing terminologies by hand would be difficult and time-consuming. Thus, we would like to automate this process as much as possible. A partially automated solution to this problem is possible whenever another two-level ontology already exists for the given domain. We call the process of generating an upper level for a terminology by using an existing two-level ontology in the same domain *semantic enrichment*. In this paper, we present an application of semantic enrichment to two preexisting medical terminologies.

In Section 2 we will discuss in more detail what the perceived advantages of the two-level structure of the UMLS are. In Section 3 we introduce our method of semantic enrichment. Section 4 describes our architecture and implementation. Section 5 contains experimental results. Related work is discussed in Section 6. Section 7 contains our conclusions.

## 2. Advantages of the two-level structure of the UMLS

Researchers at the National Library of Medicine, the "owners" and implementers of the UMLS, write about the advantages of the two-level structure [13]:

> "A two-level approach allows for organizing a small, stable, high-level taxonomy for subsequent use in

[3] http://www.nlm.gov/pubs/factsheets/umlsmeta.html.

reasoning activities. On the other hand, it allows for classifying the huge amount of lower-level concepts so that the most specific applicable knowledge can be inherited from the upper-level taxonomy."

In this Section, we will discuss advantages that go beyond those mentioned in the above quote. In Section 2.1 we will discuss two well-established advantages of the two-level structure. These ideas will be further elaborated in Section 2.4. Sections 2.2 and 2.3 describe hypothesized advantages that we have found plausible, however, more studies are necessary to confirm those advantages. Section 2.5 introduces the issue of generating two-level knowledge bases from one-level knowledge-bases.

### 2.1. Semantic types help with integration and auditing

The UMLS itself was built by integrating more and more terminologies into the existing Metathesaurus. In 1989, the version known as META-1 (which was not the first version) contained on the order of 60,000 concepts, derived from MeSH, SNOMED, ICD-9-CM, CPT-4, DSM-III, etc. [7]. By 1995, a core of merging techniques had been developed [15]. Integrating a new terminology into a Metathesaurus of several hundred thousands of concepts is extremely difficult, even with computational tools. By first classifying the concepts of a new terminology using the Semantic Network, this task becomes more manageable. A new concept does not have to be compared with every concept in the Metathesaurus, but only with those UMLS concepts that have the same assignments to semantic types as the new concept. This considerably reduces the difficulty of integration, as long as all assignments to semantic types have been made correctly and consistently. Thus, the two-level structure has supported the integration of new terminologies, and with that the creation of the UMLS itself.

Previous studies have explored the advantages of the two-level structure of the UMLS Semantic Network for supporting integration and interoperability among available resources [13,16]. In Bodenreider et al. [16], prior categorizations (semantic types) of the original and the target concepts were used to prevent irrelevant mappings. For example, we do not want a match of *merlin* (a gene product) to *Falco colombarius* (a bird also called *merlin*). Similarly, *oxygen sensor* (a molecular function) should not be matched with *oxygen sensor* (a medical device). Pisanelli et al. [17] mentioned that the UMLS two-level structure provides invaluable advantages in ontological analysis and integration without multiplying the ad hoc distinctions.

We have also done work on the problem of ontology integration based on a two-level ontology [18]. Our approach to ontology integration simplifies the matching task by identifying sets of semantically similar concepts before starting with the actual matching steps. In [18], we only compared terms from two ontologies during integration, if they were already classified as semantically similar according to the UMLS. We showed that the two-level ontology reduced the likelihood of false positives. In ontology integration, "false positive" means that two terms from two ontologies are reported by the integration algorithm to stand for the same concept, even though they are different. The desirable effect of avoiding false positives is achieved, since we avoided even trying to match concepts of different semantics, which out of principle cannot be the same, as long as all semantic type assignments are correct. Using the two-level ontology leads to better precision, with limited deterioration of recall. Our methodology has an additional advantage. It reduced the computational cost of the matching operations. Fewer pairs of terms had to be matched against each other. Specifically, the run time was about 30 times faster for the two-level ontology than for a control case. For more details on our previous work on two-level ontologies see [1,19].

Another application area, where two-level structures are helpful, is auditing of terminologies. Auditing of terminologies is defined as the principle-based computer-supported detection of mistakes in terminologies. We have shown in previous work [1] that the two-level structure makes it easier to find mistakes in terminologies. It becomes possible to develop novel methods for auditing an ontology that has been built as a Terminological Knowledge Base [1]. In brief, the two-level structure allows the creation of "intersection types" which result in (relatively) small uniform sets of concepts. Intersection types will be discussed in more detail below, especially in Section 2.4. It was shown that it is comparatively easier for a human to find omissions, wrong categorizations, etc. in such uniform sets [1]. Certain mistakes, called redundant categorizations, can be found algorithmically [20].

### 2.2. Coarse distinctions may be easier to make than fine distinctions

The backbone of most ontologies is a concept hierarchy or taxonomy. When constructing such a hierarchy, a knowledge engineer, with the help of a domain expert, needs to assign concepts from a given list of concepts to locations in the hierarchy. This hierarchy typically is a Directed Acyclic Graph (DAG). The knowledge engineer might be helped by a variant of the classification algorithm [21], however, finding exact placements of concepts in the hierarchy is still difficult with real world knowledge.

One of the accepted rules of knowledge engineering is that when a new concept has to be added to a given taxonomy, it needs to be added under the most specific concept available that still subsumes it. For example, a taxonomy may contain the following path: Living_ Thing -> Plant -> Flower -> Daffodil. Adding Rose under Daffodil would be wrong, because a Rose is not a Daffodil. Adding Rose under Plant would be correct, but it is not the most specific concept available. In this example, the only correct place to add Rose is under Flower, as a sibling of Daffodil.

The knowledge engineer needs to make a large number of such decisions. But when working at low-levels of the hierarchy, the knowledge engineer will be forced to make finer and finer distinctions. Experience and anecdotal evidence show that making these fine distinctions is very difficult and, when there is a team involved, it is often the source of long discussions.

On the other hand, assigning a concept to a broad category is almost always easier, unless there are coarse categories with a high degree of overlap. The two-level structure of a TKB allows a knowledge engineer to constrain the semantics of a concept by assigning it to one or a few semantic types (broad categories). Given the high level nature of semantic types, those assignments will typically be uncontested, or at least cause fewer disagreements as will be shown below with an example. We hypothesize that making coarse distinctions is easier than making fine distinctions and will present a few examples for this from different domains. For example, a diode is an electronic component. A gear wheel is a mechanical component. These broad categorizations would not be sources of great dissent or extended discussions. But is a diode an active or a passive component? Such finer distinctions are much harder to make. To switch to another domain, is a cat an animal or a plant? Is a tulip a flower or a tree? Again, the determination of these classifications is easy, because animal, plant, flower, tree, etc. are very broad categories. On the other hand, do you know whether a "Tubergen's Gem" is a Tulipa Clusiana (a class of tulips) or a Tulipa Chrysantha (another class of tulips)? Very detailed expert knowledge is necessary to make such a fine determination, but most people would recognize (e.g., on a picture) that a Tubergen's Gem is a tulip (or at least, that it is a flower).

In intuitive terms, to be elaborated on later in the paper, in our approach experts assign concepts to one or several *coarse* categories. If two experts assign the same concept to two different categories, we assume that they are both correct, *as they are experts*. By allowing a combination of coarse categories we achieve finer shades of semantics without the more difficult task of assigning a concept to a fine category. We now have concepts belonging only to the first category, concepts belonging only to the second category, and concepts belonging to both categories. In other words, the assignment to a fine category is *computed* based on the human assignments to coarse categories. This assumes that experts will rarely ever make an assignment that is outright wrong, an assumption that needs to be verified by future studies. The efficacy of the process of computing "intersection types" was established in [1].

If experts could assign concepts only to one of $N$ existing semantic types, then there would be only $N$ kinds of semantics possible. However, if experts may assign concepts two semantic types, then there are already $N*(N-1)+N$ kinds of semantics possible. With three and four semantic types the number of possible shades of meaning increases even further. To find how many such different "shades of semantics" might occur in an ontology is a project in itself, beyond the scope of this paper. However, in work on the UMLS Semantic Network itself it was found that allowing multiple assignments increases the number of shades of semantics by about an order of magnitude [23]. To summarize, we hypothesize that the two-level structure is helpful, because it relies heavily on coarse classifications. These are presumably easier to make than fine distinctions. Yet, by allowing several coarse distinctions for one concept, the resulting combination of semantic types may define quite a fine distinction.

### 2.3. Natural semantic types are easier to use in a two-level structure

Another hypothesized advantage of the two-level structure is rooted in the choice of the semantic types themselves. Looking at the UMLS, one finds that most of the semantic types such as Animal, Virus, Bacterium, Mammal, Human, Plant, Event, Injury or Poisoning, Organism Function, Mental Process, Environmental Effect of Humans, etc. appear natural to any medical expert. That means that they can be understood without any additional explanation. However, we will now show examples that make us conclude that natural semantic types are more difficult to find when constructing one-level hierarchies.

We will now use the common definition that the root of a taxonomy is at level 1, its children are at level 2, its grandchildren are at level 3, etc. We observe the following about a well-known *one-level ontology*. Referring to [http://www.cyc.com/cycdoc/upperont-diagram.html], OpenCYC contains at level 3 PartiallyIntangibleIndividual as a child of PartiallyIntangible and Individual. At level 5 we find the concept PartiallyTangible. We would assume that PartiallyTangible and PartiallyIntangible better be at the same level. Alternatively, if we allow for things to be Tangible or Intangible (does*tertium non datur* apply?) then would not PartiallyTangible and PartiallyIntangible objects have the same extension? This is one isolated example, and, no doubt, a discussion with the designers of CYC would reveal good explanations for this structure. Yet, a knowledge representation is supposed to stand for itself and should not need extended explanations. The knowledge representation *should be the explanation*. As we are sure that this structure was well thought out, we hypothesize that its unnatural "look" is the result of limitations imposed by a one-level hierarchy.

Other ontologies [22] fare not much better in our opinion. The Generalized Upper Model (GUM) combines "Saying and Sensing" together, as well as "Being and Having" and "Doing and Happening" which are all three children of…

… Configuration. What does Saying have in common with Sensing? Does Shouting or Signaling with Sign Language (if they occur in the GUM at all) have more in common with Saying than Sensing? Intuitively, we would assume that a Configuration is, for example, a list, a set, a bag, a graph, or alternatively a triangle, pentagon, etc. If all children of Configuration are grammatically in gerund form, than should not Configuration at least be "Configuring"? The bottom line is that it is again very hard for the uninitiated to understand what exactly is or is not under Configuration, thus defeating the purpose of representing something akin to everyday knowledge. Once again, our purpose is *not* to criticize CYC, GUM and other ontologies. Rather, the designers of those ontologies have been forced into creating unnatural concepts by the limitations forced upon them by a one-level knowledge structure.

Apparently, when building a one-level hierarchical structure, the designers are under pressure to account for everything at every level of the hierarchy. In other words, a few high level concepts have to cover all the lower level concepts under them. If no such powerful concepts exist for the higher levels, then they have to be invented, which makes them by definition unnatural. By allowing concepts in a two-level structure to point to any semantic types, bypassing their own concept parents at will, there is less of a need to introduce artificial concepts such as Configuration or PartiallyIntangibleIndividual which are not intuitive and require further explanation. We consider this to be a constructive insight. Building Terminological Knowledge Bases will only be successful if the semantic types used are intuitive and natural. We hypothesize that it is easier to classify concepts by using a few well-understood semantic types as opposed to many badly understood artificial concepts even if the latter are the result of an analysis of great depth and intelligence.

Formally defining and measuring "naturalness" is difficult. However, when a word has to be invented for a concept, such as PartiallyIntangibleIndividual then it is of low naturalness. Similarly, a combined semantic type with no assigned concepts would get a low score of naturalness.

### *2.4. Good combinations of semantic types*

In some cases, experts agree that certain entities genuinely belong to two semantic types. Thus, a loudspeaker is an electric component. A loudspeaker is also a mechanical component. Therefore, a loudspeaker has to be classified as both. Loudspeakers are not the only components that have both electrical and mechanical properties. Electric engines, relays, microphones, conventional vibration transducers, etc. all combine electrical and mechanical features. Experts realized this a long time ago, so they invented a whole new category of electromechanical devices to categorize them. The concept that is the result of the combination of the concepts electrical device and mechanical device describes the *intersection* of electrical and mechanical devices.

There is a lesson to be learned from this, too. In many cases there exist combinations of semantic types that co-occur with some regularity. One may classify all concepts that belong to such a combination of semantic types as belonging together. By highlighting the fact that there are indeed many concepts belonging to a specific combination of semantic types, a new level of clarity is gained.

In previous work [1], we have extensively studied the conditions and effects of introducing new semantic types that combine groups of frequently co-occurring "old" semantic types. We called such a semantic type an *intersection type* [1] in Section 2.1. The introduction of intersection types is a contribution of our own research team to the study of the UMLS. We have greatly systematized the process of creating intersections (such as "electromechanical") and have created a new Refined Semantic Network (RSN) with a much cleaner structure than the original Semantic Network of the UMLS [23]. In the RSN, every concept is assigned to one single semantic type only.

Let us clarify how the two-level approach is superior to the traditional approach, using an artificial example similar to the one above. Let us say that at some point in history the category electromechanical devices did not exist yet. Then the first electromechanical device, say a motor, was invented. Let us further assume that one expert classifies this device as Electrical System. Another expert classifies the device as Mechanical System. Even if the two experts never talk to each other, our system is able to *algorithmically* create a new category of things that are both electrical and mechanical systems. If desired, our system may even generate a name for this category, although it would not be a nice name, just a concatenation of the two existing semantic type names.

At this point ontology designers have a great deal of flexibility. They may *take the liberty to never revisit their original classifications*, thereby accepting the algorithmically created new category by default. They may purposefully decide that an intersection type with just one concept is too insignificant to even deserve a name of its own, again accepting the algorithmically created category. Or they may decide to use the algorithmically created concatenated name. This would be reasonable when the English language does not contain a "natural" term to describe the intersection. Alternatively, they may decide that there is indeed a "natural" term to describe an intersection. Thus, in previous research we encountered the intersection of Body Part and Mechanical Device, which we then realized is a Prosthesis. Lastly, the experts may wait till several concepts have come into being that all fit into the intersection of electrical and mechanical devices. At that point the original experts might be alerted (*automatically, by the system)* to review those concepts and to come up with a name for this intersection. Thus, our approach introduces a new set of options how to assign semantics to concepts. The goodness of an intersection type may be approximated by measuring how many links it eliminates. Thus, if there are $N$ concepts which are all assigned to the same $k$ semantic types, and then one new intersection type is introduced, this will eliminate $k * N$ concept assignment links, add $N$ concept assignment links to the new intersection type, and add $k$ new IS–A links, for a total change of $C = -k * N + N + k$. The larger the absolute value of $C$ is, the more useful is the newly introduced intersection type.

### 2.5. The need for generating two-level ontologies

In the previous subsections, we have discussed advantages of two-level ontologies, both advantages supported by our previous research and hypothesized advantages which need further investigation. Unfortunately, there are many important ontologies "out there" on the Web, which are not structured as Terminological Knowledge Bases. In this paper, we are addressing the question of how to transform a one-level terminology into a two-level TKB. Due to the great difficulty of this problem, we are only addressing the case where a global ontology exists in the same domain as the given one-level terminology. Thus, we are showing in this paper how to build a TKB out of a one-level ontology by finding its concepts in a global ontology that has semantic types (two levels).

The process of finding semantic types for concepts is difficult, even in the medical domain, with the UMLS readily available. In a relevant study [12], discuss an integration problem which requires the assignment of concepts to semantic types as a subproblem. The authors indicate that the assignment of concepts to the UMLS semantic types was done by hand. The authors write: "A default STY's assignment, according to the intended meaning of the MST table titles, proved not to be useful since there is a huge amount of heterogeneity within the tables." We do not purport to offer a fully algorithmic solution either. However, our method, called semantic enrichment, should offer a way to advance the state of the art and to move the ratio of human effort to computer effort closer to the desired value of 0.

While most of our work is done with medical terminologies, the principles developed here are of a general nature. As noted before, our work concentrates on the UMLS. However, in the future we are planning to use the WordNet/SUMO combination for additional research. Concerns that our approach does not generalize from the medical domain appear unfounded. For example, [24] showed that there is a strong compatibility between medical and nonmedical ontologies. One fifth of the UMLS semantic types had exact mapping to the standard Upper CYC Ontology and 48% of the UMLS semantic types have matches in WordNet.

## 3. Semantic enrichment

### 3.1. Basic definitions

In our previous work [18], an investigation of the formal basis of semantic enrichment is presented. We now discuss a formal treatment of semantic enrichment.

**Definition 1** (*Terminological Knowledge Base*). We call any structure that consists of (1) a semantic network of semantic types; (2) a thesaurus of concepts; and (3) assignments of every concept to at least one semantic type, a Terminological Knowledge Base (TKB).

Thus, a TKB is a triple:

$$\mathrm{TKB} = \langle \hat{C}, \hat{S}, \mu \rangle \tag{1}$$

in which $\hat{C}$ is a set of concepts, $\hat{S}$ is a set of semantic types, and $\mu$ is a set of assignments of concepts to semantic types. We will use capital letters for semantic types and small letters for concepts.[4] Finally, $\mu$ consists of pairs ($c$, $S$) such that the concept $c$ is assigned to the semantic type $S$.

$$\hat{S} = \{W, X, Y, \ldots\}; \ \hat{C} = \{a, b, c, d, e, \ldots\}, \tag{2}$$

$$\mu \subset \{(c, S) | c \in \hat{C} \& S \in \hat{S}\}. \tag{3}$$

In [1], Ŝ and Ĉ formed separate DAG structures. We will discuss structural constraints on these sets below. Furthermore, it holds:

$$\forall c \in \hat{C}[\exists\, S \in \hat{S}[\exists\, p \in \mu[p = (c, S)]]]. \tag{4}$$

In words, every concept must be assigned to at least one semantic type. The opposite condition does not hold.

In many situations there is no two-level structure available. To create a two-level ontology we propose the following naive approach: For every concept in a one-level (local) terminology, check the bottom-level of a two-level (global) ontology in the same domain and find the concept there. Then assign to the local concept its corresponding global semantic type from the top-level. Done. In the medical domain, the two-level ontology would be the UMLS.

Clearly, the naive approach, using the UMLS, would only work in the medical domain. But, because of the enormous size and wide coverage of the medical field by the UMLS, the naive approach should be easy to perform. We attempted to use the naive approach with two small medical terminologies which will be described below. The initial experiment ended up as a surprising failure. In response to this failure, we collected and analyzed cases where a human was able to find a semantic type for a concept, but the naive algorithm was not. We will describe the results of this analysis below. First, we briefly survey the two medical terminologies that we used. The American College of Cardiology (ACC) has provided a list of 142 terms with definitions [http://www.acc.org]. These concepts are separated into 22 "categories." The Society of Thoracic Surgery (STS) has created a classification of 248 terms, subdivided into 21 categories [http://www.sts.org/]. Regrettably, the categories are not always assigned consistently. Furthermore, in many cases the categories are *not* generalizations of the terms. Thus, the optimistic assumption that "term IS–A category" holds, cannot be made. For example, in many cases a term describes an attribute of a category. Therefore, neither the ACC nor the STS qualify as TDKs. We will go into more details on this problem in the next section.

[4] Both Roman and italic fonts.

### 3.2. Difficulties in handling medical terminologies

Now we will describe some obstacles that we have encountered during the semantic enrichment of the STS and ACC ontologies. Both these ontologies have concepts and categories for describing cardiovascular domain knowledge. Two major issues we faced were: (1) a great degree of inconsistency exists among the (perceived) relationships between concepts and categories; (2) inconsistent patterns appeared in either the concept names or the category names. The inconsistent naming created major obstacles in matching and automated categorization. Above we wrote "perceived relationships," because the ontology itself does not name the relationship that is supposed to hold between one concept and its category. Thus, the user is left with the task of guessing each relationship.

Intuitively, the categories classify concepts in the STS and ACC (similar to semantic types in the UMLS). This is what the name "categories" seems to imply. However, we can only treat something as a semantic type if it *is used like a semantic type.* We firmly subscribe to Wittgenstein's principle of "meaning is use." Semantic types are used to classify concepts. Thus, the relationship between a concept and a category must be one of classification, otherwise the category does not qualify as a semantic type.

As mentioned before, there exist several different kinds of relationship between concepts and categories of the ACC and STS. Not all of them are equivalent to classification. This forces us to evaluate each relationship and to incorporate its treatment in the semantic enrichment algorithm. If there exists an IS–A relationship between a concept and a category, then a semantic type of the category can be propagated to the concept. Otherwise, the use of the category information provided by the ACC and the STS depends entirely on the nature of the relationship that is presumably holding between the concept and the category. One fact is sure: propagation typically does not lead to correct results. In short, categories are not semantic types, because they cannot be used as semantic types.

In Table 1, each IS–A relationship describes a super/subclass relationship between a concept and a category (e.g., Gender is a Demographics [Item]). In the ACC and STS ontologies, the category occasionally appears as prefix or postfix in the concept name. Those prefixes or postfixes provide additional context, which is useful for determining the semantic type of a concept (e.g., Thrombolysis-Intvl contains Intervention as a postfix and RF-Diabetes contains Risk Factor as a prefix). Thus, we define IS–A (Prefix-of) and IS–A (Postfix-of) relationships as IS–A relationships. Occasionally, like above, a prefix or postfix occurs as an abbreviation. However, this does not have to be the case. To handle acronyms, a list of domain specific acronyms can be stored in a database and converted into full names such as Risk Factor for RF, Medications for Meds, Valve Surgery for VS and Vessel Disease for VD.

The Attribute-of relationship describes that a concept is a database field of a category (e.g., Participant ID is a field of the Administrative table). The Instance-of relationship defines a concept as a specific instance of a category (e.g., Comps-Neuro-Cont Coma ⩾ 24 h is an instance of Complications). There are some *ambiguous* categorizations that exhibit a lack of evidence for determining a concept as belonging to a category (e.g., Hypertension is a category of History and Risk Factors, Diabetes is a category of History and Risk Factors).

Table 2 shows some patterns that appeared in ACC or STS concepts. The Instance-of relationship describes a relationship between words in the concept (e.g., Pulmonic valve disease is an instance of Valve Disease). In the multi-word case of the form Skin Incision Start Time the last word Time determines the semantic type while in the case of Primary Cause of Death, the word Cause before of determines the semantic type. In a noun–noun phrase, the determining word is typically the second noun, which is referred to by linguists as *head noun.* However, there are famous exceptions to this rule, such as *toy gun*, which is a toy, not a gun. In this case, the first noun would be used to determine the semantic type of the noun–noun phrase.

The string "(min)" is marked as *redundant*, as it is not really a part of the concept term, but provides additional information about this concept. In this specific case, it provides the unit of measurement of the quantity that is measured by the concept.

Table 1
Examples of relationships between concepts and categories in ACC/STS

| Relationship | Concept | Category |
|---|---|---|
| IS–A | Gender | Demographics |
| | Weight | History and Risk Factors |
| IS–A (prefix-of) | RF-Diabetes | History and Risk Factors |
| | Meds-Digitalis | Pre-Operative Medications |
| IS–A (postfix-of) | Thrombolysis-Intvl | Previous Interventions |
| | Ace-Inhibitors—Discharge | Discharge |
| Attribute-of | Participant ID | Administrative |
| | Hospital ZIP Code | Hospitalization |
| Attribute-of (compound) | Patient SSN/Country Code | Hospitalization |
| | Clopidogrel/Ticlopidine | Medications |
| Instance-of | Left Main Dis >50% | Diagnostic cath procedure-findings |
| | Comps-Neuro-Cont Coma ⩾24 h | Complications |

Table 2
Complications that appeared in ACC/STS concept names

| Pattern | Name | Description |
|---|---|---|
| Instance-of | Valve disease—Pulmonic | Pulmonic valve disease is an instance of *Valve Disease* (Indicate whether there is evidence of regurgitation of the pulmonic valve) |
| Acronym-of | VS-Aortic Proc-Procedure | VS is a Valve Surgery. Proc is a Procedure |
| | VD-Insuff-Mitral | VD is Vessel Disease |
| Synonym-of | Patient DOB | DOB is Date of Birth |
| Multi-words | Conversion to Std Incision | Conversion determines the semantic type |
| | Skin Incision Start Time | Time determines the semantic type |
| | Primary Cause of Death | Cause determines the semantic type |
| Redundant word | Cross Clamp Time (min) | (min) is redundant |
| | Unique Patient ID | Unique is redundant |
| Symbol CAB | During This Admission—Date | "-" is a symbol |
| Abbreviation | Comps-Op-ReOp Other Card | Comps is an abbreviation of Complications |
| Compound words | Comps-Op-ReOp Bleed/Tamponade | Bleed and Tamponade are compound words |
| Inconsistency | ⩾ and "Greater than Equal" | Different notations for the same concept |

### 3.3. Basic definitions of semantic enrichment

In this section, we will formally describe semantic enrichment. In the definition of TKBs we did not specify what structures may be formed by the concepts of the lower level or by the semantic types of the upper level. The structure of the ACC and of the STS is much simpler than what is commonly found in ontologies, and we limit ourselves to this kind of structure.

**Definition 2** (*Local Ontology*)**.** A local ontology is a one-level ontology that consists of concepts and categories. Each concept is associated with one category.

In the definition we used the term "associated with," because the exact nature of the relationship between a concept and a category is not fixed, as was shown in several previous examples from the ACC and the STS. We note that the local ontologies are considered to be fundamentally different from the global ontology, which is based on observing that the sample ontologies used in our research are indeed fundamentally different from the UMLS (see Section 3.2). Thus, Definition 2 as distinct from Definition 1 is needed.

**Definition 3** (*Global Ontology*)**.** A global ontology is a TKB in which the upper level is organized as an IS–A hierarchy of semantic types. Its lower level is an IS–A hierarchy of concepts and exhibits wide and deep coverage of the concepts of the domain for which this ontology is defined. The properties of the global ontology have been defined in TKB (1)–(4).

**Definition 4** (*Semantic Enrichment*)**.** Semantic enrichment is any process that takes as input a local ontology and a global ontology and produces as output a TKB that has in its lower level the same concepts as the local ontology and these concepts are assigned to semantic types from the global ontology.

*Note.* Even though the categories of the local ontology are used to perform the semantic enrichment operation, they are not considered part of the final resulting TKB.

The semantic enrichment process is composed of three steps: (1) concept matching, (2) semantic assignment and (3) assignment propagation.

**Definition 5** (*Concept Matching*)**.** Concept matching is a process which either finds for a concept ($c_l$) of a local ontology a corresponding concept ($c_g$) from a global ontology or for a category ($a_l$) of a local ontology a corresponding concept ($c_g$) from a global ontology. The result is a pair ($c_l, c\ _g$) or ($a_l, c_g$): furthermore, for the second kind of pair we define a mapping function Match such that Match ($a_l$) = $c_g$.

Two concepts (or a category and a concept) are considered *corresponding* when they are identical according to a suitable string match, or similar enough as strings to warrant the assumption that they stand for the same real world (abstract or concrete) entity. We will return to this issue in Section 3.5. Fig. 2A shows the step of concept matching. In this step an attempt is made to match the local concept $c_i$ against any concept in the global ontology. For the purposes of this example we assume that this attempt is successful. Thus, $c_i$ matches $c_{g1}$.

Because the concept $c_i$ is connected by an IS–A link to the category $a_i$, we also attempt to match $a_i$ against the concepts of the global ontology. In the given example this attempt is also successful, and therefore we get a match of $a_i$ with $c_{g2}$. In Fig. 2A, this IS–A link is marked by an arrow from $c_i$ to $a_{i.}$

**Definition 6** (*Semantic Assignment*)**.** Semantic assignment is a process which creates a pair of a concept ($c_l$) or category ($a_l$) from the local ontology and a semantic type $S_l$, which is a copy of a semantic type $S_g$ from the global ontology, such that $S_g$ is the semantic type of the $c_g$ that was found during Concept Matching: ($c_l$, $S_l$) or ($a_l$, $S_l$).

In practical terms, this corresponds to a step that we are performing while constructing the upper level of the TKB. If a copy $S_l$ of $S_g$ of $c_g$ already exists in the upper level of the new TKB, then the assignment of $c_l$ to $S_l$ is immediately added to $\mu_l$. Otherwise, a copy of $S_g$ has to be made and

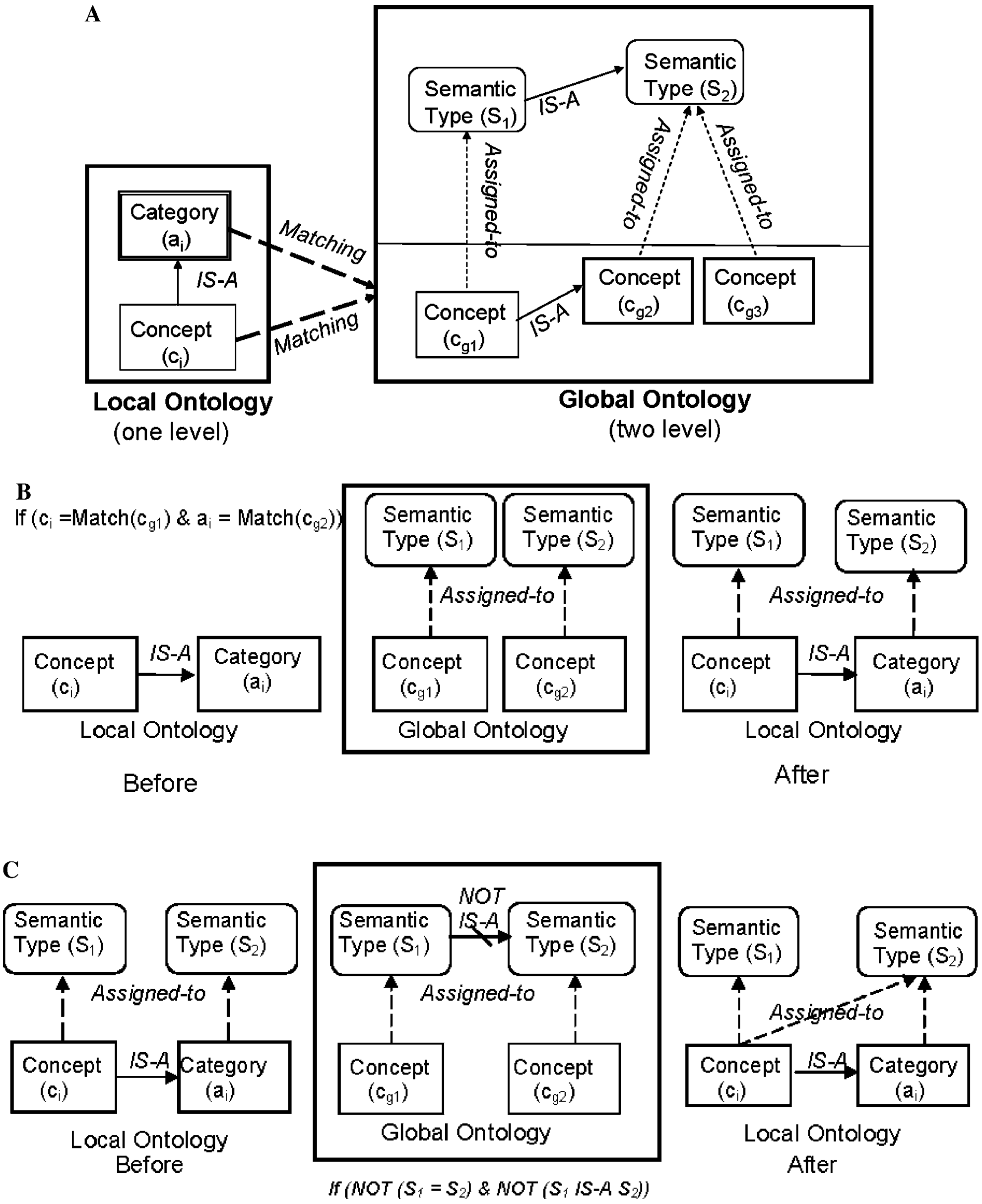

Fig. 2. (A) Step 1: concept matching. (B) Semantic assignment and (C) assignment propagation.

placed in the new TKB. Afterwards $(c_l, S_l)$ or $(a_l, S_l)$ is added to $\mu_l$. Fig. 2B shows the step of semantic assignment in the middle. We have found matches for both $c_i$ and $a_i$. Therefore, we create copies of both $S_1$ and $S_2$ in the local ontology. We also add assignments of $c_i$ to $S_1$ and of $a_i$ to $S_2$ into $\mu_l$, the local assignment set that we are constructing.

If semantic assignment is performed for a concept $c_l$, then this is the "obvious" case.

However, in our experience with the ACC and the STS, it has been impossible to perform a semantic assignment for many concepts. Therefore, we attempt to find a semantic type for a concept indirectly in a two-step process. First, we perform a semantic assignment of a semantic type to a category $a_l$. Then we perform assignment propagation, defined below, to the concept.

**Definition 7** (*Assignment Propagation*)**.** Assignment propagation is a process that "inherits" a semantic type $S_l$ from a local category $a_l$ to a local concept $c_l$, provided that an IS–A relationship holds between $c_l$ and $a_l$, and provided that $S_l$ was assigned to $a_l$ during a step of semantic assignment.

Assignment propagation is sensible for the following reason. In some cases, concepts are ambiguous. However, the category may eliminate or reduce this ambiguity. For example, "cold" may be the disease "common cold" or a statement of temperature. (COLD may even stand for Chronic Obstructive Lung Disease). If "cold" is assigned to the category "disease," this ambiguity is eliminated. Thus, the semantic type of the category should help to better define the meaning of the concept. This assumes, however, as noted above, that the concept and category really stand in an IS–A relationships to each other, which was not always the case.

If several semantic types have been assigned to *al*, then all of them will be "inherited" to $c_l$. Formally speaking, for a single semantic type $S_g$ and its local copy $S_l$ the following holds:

$$\text{IS–A}(c_l,a_l)\&(\text{Match}(a_l),S_g)\in\mu_g\rightarrow(c_l,\text{Copy}(S_g))\in\mu_l. \quad (5)$$

Fig. 2C shows in Step 3 how assignment propagation is performed. We assume (see "Local Ontology Before") that the category $a_i$ has acquired the semantic type $S_2$. Because there is no IS–A link from $S_1$ to $S_2$ (and no IS–A link from $S_2$ to $S_1$) $S_2$ qualifies as a valid semantic type for $c_i$ also. We stress that the NOT IS–A "link" from $S_1$ to $S_2$ is not a link that is named NOT IS–A. Rather, this is an explicit representation that no such link exists.

Normally, we would not mark the absence of a link. However, in this example, this absence is crucial, so we make it explicit. Because there is no IS–A link from $S_1$ to $S_2$, the assignment link from $a_i$ to $S_2$ is propagated to become an additional assignment link from $c_i$ to $S_2$.

In the example in Fig. 2, both $c_i$ and $a_i$ had matches in the global ontology. However, in many cases only the category ($a_i$) has a match, and thus the use of assignment propagation is the only way to find a local semantic type for $c_i$.

### 3.4. Prohibited propagations

Not every propagation is permissible. We will discuss two cases when propagation may not be performed, called assignment propagation prohibition and assignment propagation redundancy.

**Definition 8** (*Assignment Propagation Prohibition*). Assume that a concept $c_l$ is connected to a category $a_l$, and a semantic type $S_l$ has been assigned to $a_l$ by copying $S_g$ from a global ontology. If the connection from $c_l$ to $a_l$ is neither an IS–A nor an Instance-Of link, then the propagation of $S_g$ to $c_l$ is prohibited.

In the ACC and STS, the only major kind of connection between $c_l$ and $a_l$ for which assignment propagation prohibition applies is the Attribute-of connection. However, in other domains more such connections may exist.

Formally speaking,

$$\begin{aligned}&\text{NOT}(\text{IS–A}(c_l,a_l)\text{OR Instance}\\&\quad-\text{Of}(c_l,a_l))\text{OR NOT}((\text{Match}(a_l),S_g)\\&\quad\in\mu_g)\rightarrow\text{NOT}((c_l,\text{Copy}(S_g))\in\mu_l).\end{aligned} \quad (6)$$

**Definition 9** (*Assignment Redundancy*). An assignment of a concept $c_l$ to a semantic type $S_1$ is redundant if and only if $c_l$ is also assigned to a semantic type $S_2$ and $S_1$ is a parent or ancestor of $S_2$ in the global TKB.

**Definition 10** (*Propagation Redundancy*). A propagation of a semantic type $S_l$ from a category $a_l$ to a concept $c_l$, which is possible whenever $c_l$ IS–A $a_l$ (or $c_l$ Instance-of $a_l$), is redundant if $c_l$ is already assigned to the semantic type $S_l$.

In Fig. 3, to demonstrate an example of propagation redundancy, the ACC concept Aspirin is assigned to two semantic types Organic Chemical and Pharmacologic Substance. Aspirin's category, Medications, is also assigned to Pharmacologic Substance. The relationship between the concept Aspirin and the category Medications is IS–A. Because Aspirin is already assigned to the semantic type Pharmacologic Substance, it is not necessary to propagate Pharmacologic Substance along the IS–A relationship from Medications to Aspirin. Based on the previous two definitions, we can introduce a new definition.

**Definition 11** (*Assignment Propagation Redundancy*). A propagation of a semantic type $S_1$ from a category $a_l$ to a concept $c_l$, such that $c_l$ IS–A $a_l$, is redundant if $c_l$ is already assigned to a semantic type $S_2$ and $S_1$ is a parent or ancestor of $S_2$ in the global TKB.

The example in Fig. 4A shows an example of assignment propagation redundancy. Because $S_1$ has an IS–A link to $S_2$, propagating an assignment to $c_i$ is redundant, and therefore prohibited. Thus, even though $a_i$ has a valid assignment to $S_2$, $c_i$ does *not* have an assignment to $S_2$. With the above definitions we can give a "prose" description of the process of semantic enrichment. For every concept for which semantic assignment is possible, perform semantic assignment. For every concept check whether assignment propagation to the concept from a category is possible. This requires that semantic assignment to the category is possible, that assignment propagation from the category to the concept is possible, and that neither assignment propagation prohibition nor assignment propagation redundancy precludes the assignment propagation. In Section 3.6 we will show the semantic enrichment algorithm that incorporates this description.

Fig. 4B is a valid assignment. It differs from Step 3 in Fig. 2C by the fact that there is an IS–A link from $c_{g1}$ to $c_{g2}$. Just like in Step 3 of Fig. 2 there is NO IS–A link from $S_1$ to $S_2$. As mentioned above, the absence of an IS–A is

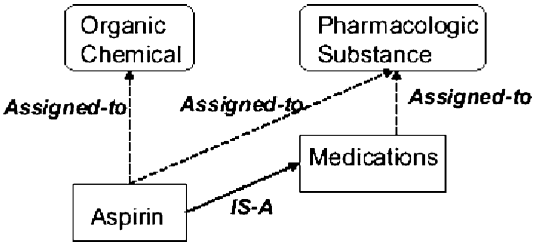

Fig. 3. An example of propagation redundancy.

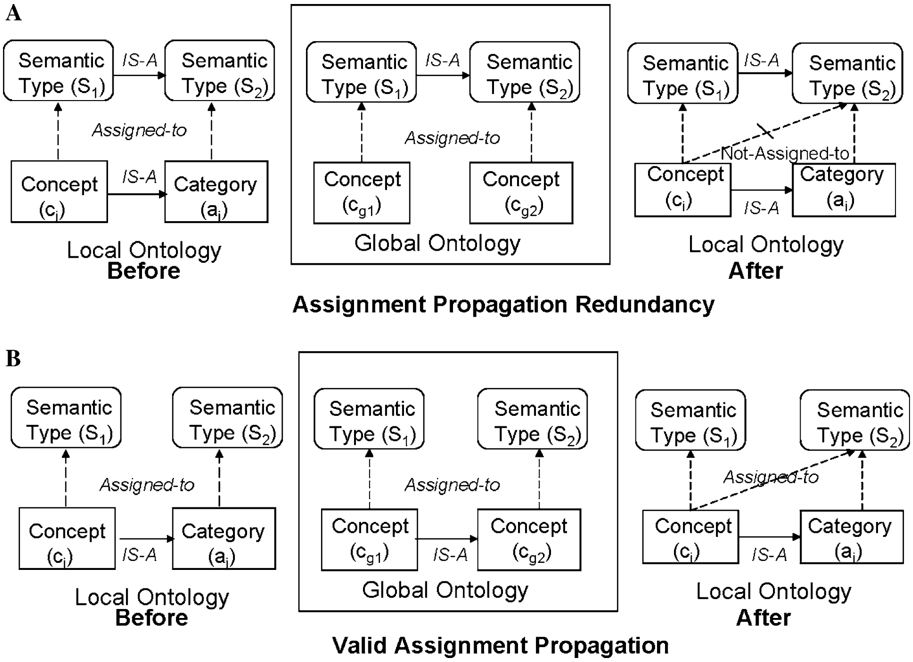

Fig. 4. Valid/invalid assignment propagation.

normally not explicitly denoted. The point of Fig. 4B is that the IS–A link at the concept level (from $c_{g1}$ to $c_{g2}$) does not block assignment propagation. Only at the semantic type level in Fig. 4A does this happen.

### 3.5. Lexical enrichment as a preprocessing step

As pointed out in the previous section, to perform semantic enrichment we need to identify pairs of concepts from different ontologies (or concept–category pairs) that have the same meaning. This step, called concept matching, requires that we overcome many issues of inconsistent naming which are usually obvious to humans but difficult to handle for algorithms. For this purpose, we use a preprocessing step called *lexical enrichment*. Lexical enrichment performs several steps, such as (1) handling acronyms or abbreviations, (2) filtering out non-alphabetic characters occurring in many medical terms, (3) deleting redundant words, (4) handling multiple or compound words, and (5) making use of synonyms and homonyms. Our solution for the lexical enrichment process is semi-automatic, meaning that in a few cases a human had to make the final judgment on a match.

First, many terminologies freely mix the use of terms with the acronyms or abbreviations of those terms. Thus, these abbreviations need to be expanded for easier concept matching. For example, the acronym RF needs to be replaced by its expansion, Risk Factor. The abbreviation “Meds.” is replaced by Medications. Other common medical acronyms in our terminologies are DOB (Date of Birth), MI (Myocardial Infarction), and many more. When an acronym occurs as a prefix or postfix (e.g., “RF-Smoker”), it is also expanded (e.g., “Risk Factor Smoker”). In this way, terms with acronyms can be matched with other terms of the same meaning.

Second, whenever terms contain special characters such as “/” or “-” they are replaced by blanks. Bleed/Tamponade is an instance of compound words containing the special character “/”. Semantically, the term Comps-Op-ReOp Bleed/Tamponade defines an operative re-intervention required for bleeding and tamponade. In this case we replace the “/” by a blank. In some cases we need to go in the opposite direction. Precise mathematical symbols are often expressed by imprecise English words in a terminology. For example, the mathematical notion “greater than equal to” is transformed from its English representation into its well defined symbolic representation “$\geqslant$”. This symbolic representation is unique, while the English representation may equally appear as “greater equal” or “greater than or equal to,” etc.

Third, there are cases where it is necessary to remove redundant or duplicate words. For example, “unique” is a redundant word in *Unique* Patient ID, because ID implicitly specifies the unique identification of a patient. Similarly, “(min)” in “Cross Clamp Time (min)” is not appropriate as part of the concept name, because it represents the unit of the given time.

Fourth, one of the challenging problems in medical databases and ontologies is dealing with multi-word terms or compound words. For example, “Conversion to Std Incision” indicates that the minimally invasive incision was converted to a full median sternotomy. This requires an analysis of the linguistic relations between the words in the term, to identify which word is most indicative of the semantic type to be assigned to the term. In this example, Conversion is the best semantic type for the multi word term "Conversion to Std Incision."

Finally, the existence of synonyms and homonyms causes problems for concept matching. The use of synonyms is absolutely necessary, because medical terminologies are full of variant terminologies (e.g., Heart and Coeur, Heart Block and Lev's disease). While acronyms can be dealt with by expansion into a canonical form, this is harder for synonyms. Rather, we have decided to include the use of synonyms during the concept matching step itself. If no match is found for a concept, then it is attempted to use its synonyms for matching.

In our implementation of lexical enrichment, the first step has been handled by referring to a domain specific acronym table describing how to expand acronyms or abbreviations into their appropriate names. The second step, filtering out of special symbols and characters, has been handled by string matching. We have processed duplicate or unnecessary words by using a table that was designed for patterns appearing in the ACC and STS terminologies. The synonyms and homonyms of terms were derived both from the STS and ACC documentations and from the UMLS.

In the case of multi-word terms, we attempt to match a concept against other concepts by using the bigram similarity approach. It is a structural approach that relies only on string similarities. The bigram approach works well when there are multi-word terms with redundancies, as those shown in Table 2. It also works well for variant grammatical forms of the same root (operate vs. operation). If the matched score for two terms is less similar than a given threshold $\alpha$ then the concept match is rejected, otherwise it is accepted. The experimental results related to the matching performance were published in our previous paper [18].

### 3.6. Algorithm for semantic enrichment

We will now present the semantic enrichment algorithm, based on the previously developed conceptualization. The preprocessing steps of lexical enrichment are not shown. We note that categories are not maintained in the final result of the algorithm, as their status is ill-defined.

Thus, all assignments of categories to semantic types are temporary and are deleted at the end of the algorithm.

```
// Input are global ontology ( $O_g$ ) and local ontology ( $O_l$ )
and output is an updated local ontology ( $O_l$ )
Algorithm: Semantic Enrichment (Ontology $O_g$, $O_l$)
    Create an empty upper level for $O_l$;
    Create an empty $\mu_l$;
    // For all the concepts in the local ontology
    FOR all $c_l \in O_l$
        // There are two mapping cases for the concepts in
        the global ontology
        FOR all $c_g \in O_g$
            // Case 1: The Local Ontology concept $c_l$ matches
                        the Global Ontology concept $c_g$.
            // Concept Matching according to Definition 5.
            IF $c_l$ matches $c_g$ THEN {
                // The semantic types of $c_g$ are assigned as $c_l$
                   semantic types.
                IF the semantic type $S_{g\ 1}$ of $c_g$ does not exist
                in the local ontology,
                THEN copy it, giving $S_{l1}$;
                // Semantic Assignment according to
                  Definition 6.
                IF $S_{l1}$ has an IS–A (Instance-of) link in the
                global ontology to any semantic type that
                exists in the local ontology,
                THEN copy the IS–A (or Instance-of) link to
                the local ontology;
                Add the assignment ($c_l$, $S_{l1}$) to $\mu_l$;
                }
            // Case 2: The category $a_l$ of the local concept $c_l$
                        matches the Global Ontology concept $c_g$
            // and between $c_l$ and $a_l$ the IS–A (or Instance-of)
                        relationship holds
            IF $a_l$ matches $c_g$ THEN {
        // The semantic types of $c_g$ are assigned as $c_l$'s
          semantic type.
        IF the semantic type $S_{g2}$ of $a_g$ does not exist in the
        local ontology,
        THEN copy it, giving $S_{l2}$ ;
        // Semantic Assignment according to Definition 6.
            IF $S_{l2}$ has an IS–A (or Instance-of) link in the
            global ontology to any semantic type
              that exists in the local ontology,
              THEN copy the IS–A (or Instance-of) link to
              the local ontology;
              Add the assignment ($a_l$, $S_{l2}$) to $\mu_l$;
             // Assignment Propagation according to
               Definition 7.
               If the assignment ($c_l$, $S_{l2}$) is not redundant
               THEN add it to $\mu_l$;
        }
    Delete all assignments of categories so semantic types
    from $\mu_l$.
```

## 4. Implementation architecture

We have implemented a Semantic enrichment prototype system [18] following the paradigm of component-oriented development [25]. The component-based development approach allows a complex system to be considered as a composition of an arbitrary number of smaller components with well-defined interfaces. Our system architecture is shown in Fig. 5. Semantic enrichment is itself only one component of a larger system for integrating ontologies. The integration issue is outside of the scope of this paper. The User Interface manager handles a user's semantic enrichment request for particular ontologies.

The TKB Builder component is composed of four subcomponents (XML Converter, XML Reader, Lexical Enrichment, and Semantic Enrichment). If it receives as

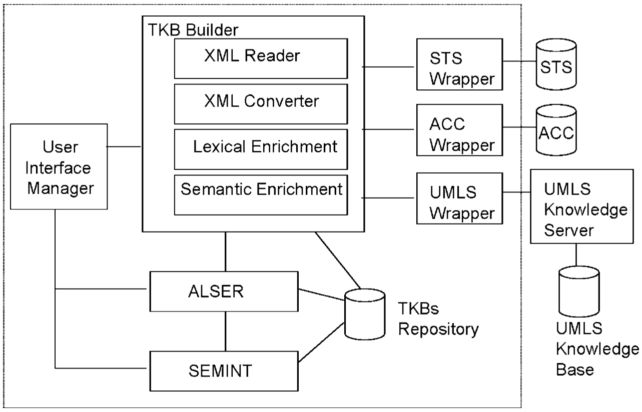

Fig. 5. The architecture.

input TKBs coded in XML, then it does not need to do anything except for passing them on. (We are using XML, because it allows us to quickly extract data and exchange information between components.) Unfortunately, most existing terminologies and ontologies are not in that format. The TKB Builder component performs the required translation of the input. If the input format is not already XML, then the input has to be transformed into XML, using the XML Converter. Specifically, the ACC [http://www.acc.org] and STS [http://www.sts.org/] terminologies and definitions were published in PDF format. Our XML Converter parses the PDF files to extract concepts and categories and then converts them into XML format. Then the XML Reader component is invoked. The XML Reader component extracts concepts and their corresponding semantic types from the XML input. The XML Reader is implemented using JAVA SAX [http://www.saxproject.org/]. The Lexical Enrichment component performs the lexical enrichment which was described in Section 3.4. This includes replacing synonyms, eliminating function words (such as articles), deleting duplicate words, expanding abbreviations and acronyms, etc.

Finally, the given terminology or ontology is transformed into a TKB, i.e., we have to perform semantic enrichment. The Semantic Enrichment component transforms terminologies into TKBs using wrappers. In our case, three wrappers are needed, the ACC Wrapper, the STS Wrapper, and the UMLS Wrapper. The ACC Wrapper and the STS Wrapper directly access their respective terminologies. The UMLS Wrapper component communicates with the Unified Medical Language System Knowledge Source (UMLSKS) server [http://umlsks4.nlm.nih.gov]. It takes concepts as input and returns corresponding UMLS semantic types.

The UMLSKS server offers several matching options. We are using "advanced search" with "approximate matching." These options were chosen to maximize the number of results. Terms from all source vocabularies in the UMLS 2003AA are used. Due to the many problems in the data that were shown in the above tables, the UMLS Wrapper was implemented as a semi-automatic task, i.e., difficult cases are processed by hand.

As a result of semantic enrichment, two Terminological Knowledge Bases were generated, encapsulating the ACC and STS terminologies, respectively. The TKBs generated by the TKB Builder are stored in the TKB Repository for future use. The ALSER and SEMINT components of the architecture are outside the scope of this paper as they are not directly related to semantic enrichment.

## 5. Experimental results

To test our algorithms, we have performed extensive experimental work in the area of semantic enrichment. Table 3 demonstrates the necessity of lexical enrichment. It shows matches of concepts in the ACC and the STS which became evident only after applying lexical enrichment to the terms in Table 2. The prefixes like "Risk Factor:" or "Value Disease:" (seen in Table 3) are used to determine the semantic type of the concepts (as described in Section 3.5). Thus, once a semantic type is assigned to a concept, the prefix will be discarded. Some concepts are rewritten, e.g., Patient DOB is replaced by Date of Birth.

Table 4 shows how STS concepts have been processed through semantic enrichment by showing their categories and semantic types. The symbol ∩ in Table 4 indicates that a concept belongs to all the semantic types connected by ∩, i.e., the intersection type. Details of intersection types are in [1]. All present enrichment processes made use of the UMLS for generating two-level ontologies by semantically enhancing the 248 concepts and 21 categories in STS and the 142 concepts and 22 categories in ACC.

Below, in Table 5, we describe the results of our analysis in quantitative terms. The first row shows the number of straight forward IS–A relationships that hold between concepts and categories. For the ACC there are only 68 out of 142, for the STS only 91 out of 248 concepts. There are also IS–A relationships that contain their category either as a prefix or as a postfix in the concept name. These add 14+3 IS–A relationships for the ACC and $38 + 8$ for the STS. Thus, in total, there are 85/142 IS–A relationships in the ACC, which comes to about 60%. For the STS there are 137/248 IS–A relationships, which comes to about 55%. Therefore, for both medical terminologies, only a little more than half of the concepts relate to their categories by an IS–A relationship. Looking again at Table 5, we can see the breakdown of how the remaining concepts relate to their categories. In the ACC, almost all remaining concepts relate to their categories as attributes. Only one concept is an instance of a category and two are "outliers" which are hard to categorize even for humans. Thus $54/142 = 38\%$ of the ACC concepts are attributes. For the STS there are 8% Instance-of relationships and 36% attributes, with only three outliers remaining. Thus the number of concepts that relate to their categories as attributes is slightly higher than one-third.

Table 6 shows that about 10% of the STS concepts match UMLS concepts exactly, and 90% match UMLS concepts with various levels of approximation. For a few

Table 3
Some examples of actual mapping before/after lexical enrichment

| Case | Original STS concept | After lexical enrichment (STS) | Original ACC Concept | After lexical enrichment (ACC) |
|---|---|---|---|---|
| Synonym | Date of Birth | Date of Birth | Patient DOB | Date of Birth |
| | Readmission reason | Readmission reason | Readmit reason | Readmission reason |
| Acronym | RF-Diabetes | Risk Factor: Diabetes | Diabetes | Diabetes |
| | VD-Insuff-Aortic | Valve Disease: Aortic | Valve Disease—Aortic | Valve Disease: Aortic |
| Redundant word | MI | Myocardial Infarction | Previous MI | Myocardial Infarction |
| | Patient ID | Patient ID | Unique Patient ID | Patient ID |
| | Payor | Payor | Insurance Payor | Payor |
| Compound word | Comps-Op-ReOp Bleed/ Tamponade | Complications: Bleeding<br>Complications: Tamponade | Vascular complications—bleeding<br>Tamponade | Complications: bleeding<br>Tamponade |

Table 4
Partial STS ontology after semantic enrichment

| Semantic type for category (after semantic enrichment) | STS category | Semantic type for concept (after semantic enrichment) | STS concept |
|---|---|---|---|
| Health care activity | Hospitalization | Temporal concept | Date of admission, Date of surgery, Date of discharge, Additional ICU hours, total hours ICU, initial ICU hours |
| Occupation or discipline | Demographics | | Patient age |
| Health care activity | Hospitalization | Health care activity* | Same day elective admission, readmission to ICU |
| Conceptual entity | History and Risk Factors | Organism attribute ∩ quantitative concept | Weight, height |
| | | Population group ∩ finding ∩ quantitative concept | Smoker |
| Health care activity | Administrative | Idea or concept | Patient ID, emergent reason |
| Therapeutic or preventive procedure | Operative | | Patient SSN, country code, urgent reason |
| Occupation or discipline | Demographics | Intellectual product | Patient last name, patient first name, patient middle initial, medical record number |
| | | Finding | Date of Birth |
| Health care activity | Diagnostic cath procedure-findings | Sign or symptom | Aortic, mitral, tricuspid, pulmonic |
| Pathologic function | Complications | | Bleeding, tamponade |
| Therapeutic or preventive procedure | Valve surgery | Therapeutic or preventive Procedure | Aortic procedure, mitral procedure, tricuspid procedure, pulmonic procedure |

Table 5
Statistics of concept–category relationships

| Relations | ACC ontology | STS ontology |
|---|---|---|
| IS–A | 68 | 91 |
| IS–A (Prefix) | 14 | 38 |
| IS–A (Postfix) | 3 | 8 |
| Attribute-of | 54 | 88 |
| Instance-of | 1 | 20 |
| Ambiguous Category | 2 | 3 |
| Total | 142 | 248 |

Table 6
Matching analysis of semantic enrichment

| Kind of match | STS ontology | | ACC ontology | |
|---|---|---|---|---|
| | Concepts | Categories | Concepts | Categories |
| Exact matches | 23 | 4 | 33 | 2 |
| Approximate matches | 221 | 5 | 109 | 2 |
| Total matches | 244 | 9 | 142 | 4 |
| Match failures | 4 | 12 | 0 | 18 |
| Total | 248 | 21 | 142 | 22 |

cases, a domain expert's involvement was required to select a semantic type. Table 7 shows the number of concept assignments for STS and ACC. There are 284 semantic type assignments even though there are only 244 concepts, because some STS concepts match UMLS concepts which have several semantic types. For 244 concepts and nine categories (Table 6) we found semantic type assignments directly, by matching them against the UMLS. For 41 concepts we found additional assignments because they inherit them from five categories. Thus, we have a total of 325 assignments of concepts to semantic types after we applied the semantic enrichment process to the STS. The corresponding numbers for the ACC are also in Table 7. Table 7 shows that 11 concepts are assigned with semantic types through assignment propagation. Table 8 shows that the three categories to 43 concepts are not considered, because the concepts relate to the categories by *Attribute-of* links. For one concept, we do not inherit the semantic type assignment due to propagation redundancy. Overall, of nine matched categories for the STS (Table 6), only five categories are used for propagation through *IS–A*. The

Table 7
Assignment and propagation analysis of semantic enrichment

| | STS ontology | ACC ontology |
|---|---|---|
| Assignment by match | 284 | 181 |
| Assignment by propagation | 41 | 11 |
| Total assignment | 325 | 192 |

Table 8
Redundancy analysis of semantic enrichment

| | STS ontology | | ACC ontology | |
|---|---|---|---|---|
| Concept category | Concept | Category | Concept | Category |
| Assignment propagation prohibition | 43 | 3 | 10 | 2 |
| Propagation redundancy | 1 | 1 | 2 | 2 |
| Total | 44 | 4 | 12 | 4 |

Table 9
Statistics after semantic enrichment

| | STS ontology | ACC ontology |
|---|---|---|
| Concepts | 244 | 142 |
| Semantic types | 38 | 35 |
| Maximum concepts assigned to a semantic type | 58 | 53 |
| Minimum concepts assigned to a semantic type | 1 | 1 |
| Average concepts assigned to a semantic type | 5 | 3 |
| Maximum semantic types assigned to a concept | 4 | 3 |
| Minimum semantic types assigned to a concept | 1 | 1 |
| Average semantic types assigned to a concept | 1.33 | 1.35 |
| Total assignments | 325 | 192 |

others are accounted for in Table 8. Table 8 also contains the numeric break down of propagation failures for the ACC. Table 9 summarizes concept assignments after semantic enrichment, separately for the STS and the ACC.

The effectiveness of the proposed enrichment methods might be measured by formal measurements such as correctness and consistency. In our work, the concepts in both ACC and STS have been processed with lexical enrichment and it turns out that there is an improvement in the semantic enrichment due to the preprocessing. Specifically, about 20% of the ACC and STS concepts have been improved through the lexical enrichment process and promising results have been obtained after the enrichment process such as 10 and 13% exact matching rates, 84 and 64% approximate matching rates and 6 and 11% matching failure rates for STS and ACC, respectively. For the approximate matching cases, a human had to make the final judgment on a match. There are multiple possibilities of semantic enrichment because different experts might make different judgments. Revisiting the presented results in qualitative terms, there is clearly room for improvement. Thus the assignment of two concepts such as Patient ID and Urgent Reason to the same semantic type, namely *Idea or Concept* may be questioned. This assignment is unintuitive and indicates that, the richness of the UMLS notwithstanding, we sometimes need better or more refined semantic types to correctly capture the meaning of concepts. However, in this study we have limited ourselves to using the UMLS semantic types. The UMLS Semantic Network is not cast in stone, and recently extensions have been proposed for it, to handle genomic concepts. Thus, if necessary, and in a very conservative way, additional semantic types may be added to a system when concept sets appear to be too heterogeneous.

## 6. Related work

The advantages of the UMLS two-level structure were well described in [26]. There it was pointed out that the two-level structure makes it possible to classify the huge number of lower-level UMLS concepts and to infer additional knowledge about them from the upper-level taxonomy. Our work is influenced by their principles of construction of the Semantic Network such as (1) we assign concepts to the most specific semantic type available, (2) we assign concepts to several semantic types, if it is necessary, and (3) we assign a concept to a less specific semantic type if no more specific semantic type is available. Pustejovsky et al. [27] linguistic work on the UMLS presented some issues which are related to our data and semantic enrichment approach.

Bodenreider et al. [16] studied the global coverage of the Gene Ontology by mapping its concepts and relations to the UMLS. They pointed out the importance of interoperability and showed that it is achievable by accessing relevant information through cross-references or similarity detection. From this interoperability perspective, in our paper a reference information source (i.e., the UMLS) was used to assign semantic types to local concepts so that related ontologies can interoperate with each other. In [12] a gastrointestinal endoscopy reporting terminology (called MTS), was integrated with the UMLS. Their mapping approach is more specific than ours, creating mappings between UMLS semantic types and MTS class attributes using inter-table and intra-table relationships of the MTS database. Thus, their available input data are much better structured than our data, which are only available to us as.pdf files. Yet, even with their better sources, they encountered many of the same data inconsistency problems that we reported on.

In Desmontils et al. [28] a semantic enrichment methodology was presented for improving an indexing process for Web pages, using terminologies like WordNet. Two types of enrichment processes are discussed: enrichment by refinement (specialization) and enrichment by abstraction (generalization). The purpose of their work is different from ours in that we focus on interoperability between ontologies by combining specialized concepts (from the local ontology) with general semantic types (from the global ontology).

We do not delete any concepts from the local ontology except when it is impossible to derive a semantic type for them by any of our methods. This is unlike the method of enrichment by abstraction [29] which deletes too specific concepts. Similar to our approach, their ontology enrichment is semi-automatic. A human expert makes the final decision whether to add a new concept to the ontology.

Gupta et al. [30] described the importance of information integration in heterogeneous biological disciplines (physiological, anatomical, biochemical, etc.) and tried to bridge the gap between heterogeneous data sources using a wrapper–mediator architecture as well as rule and F-logic based semantic integration. Their framework is still under development. Another interesting paper [31] from the same group describes the knowledge-based mediation for mapping heterogeneous resources. Their context specific language was used to specify knowledge schemas and preexisting views of global and local ontologies. Their approach to interoperability problems is similar to ours. However, their rule-based approach is different from our algorithmic approach.

Chen et al. [32] described the urgent need for semantic enrichment in e-Science on the Grid. They pointed out that the semantic Grid, enriched by an ontology, can facilitate resource sharing and interoperability on the Grid. Their solution of using semantic enrichment for task descriptions and workflows is very abstract and hard to evaluate. Colomb [33] analyzed upper level ontologies (e.g., CYC, SUMO, OntoClean, GOL, BWW, and WordNet) to support building of domain specific ontologies and finding of common ground among them, to handle semantic heterogeneity. OntoClean [34] and DOLCE [35] support semantic interoperability through reasoning engines to make it possible to interpret application-specification ontologies.

Many lines of research have addressed ontology matching in the context of ontology construction and integration [36–39]. The major goal of these approaches is to develop effective methodologies for automated mappings [40]. Work in this direction includes schema mapping methods and constraint-based semantic integrity enforcement [41], as in TSIMMIS [42], and SIMS [43].

Advanced research work in semantic interoperability includes the use of matching rules [15,38,39] and the comparison of all possible correspondences [5,44–46]. The names of concepts, the nesting relationships between concepts and the inter-relationships between concepts (slots of frames in PROMPT [39]) are also criteria for comparison. The types of the concepts, or the labeled graph structures of the models [44,47] may be used to estimate the likelihood of data instance correspondences [15,29,45,48]. Rodriguez and Egenhofer [49] proposed computing semantic similarity for different ontologies from three perspectives (1) synonym set matching, (2) semantic neighborhood, measured by the shortest path between connected concepts, and (3) distinguishing features. These three aspects are combined, using a weighted sum function.

Some similarity approaches [37,39] allow for efficient user interaction or expressive rule languages [36] for specifying mappings. Several recent publications have attempted to further automate the ontology matching process. A general heuristic was used in [50] to show that paths between matching elements tend to contain other matching elements. COMA [51] combined the similarity value of ontologies in XML and database schemas. Chimaera [37] coalesced two semantically identical terms from different ontologies and identified subsumption, disjointness, or instance relationships. LSD [46,48] developed an approach to predict available domain constraints through a learning process. GLUE [46] derived a similarity estimator to compute similarity values using the joint probability distribution between ontologies. CUPID [52] did the mapping of ontologies by using two major coefficients, the linguistic similarity and the structural similarity. In [45], similarity between two nodes was computed based on their signature vectors, which were derived from data instances. The above approaches argue for a single best universal similarity measure, whereas GLUE [46] allows for application-dependent similarity measures.

## 7. Conclusions and future work

The premise of this paper has been that ontologies and terminologies with a two-level structure have advantages over one-level ontologies. We cited extensive experience with the UMLS and recent work on WordNet with SUMO as the justification for this premise. The two level structure is independent from the way the levels themselves are organized. Thus, typically, the levels themselves contain hierarchies. The problem that we have attacked is that a majority of current terminologies are one-level structures. This paper has presented an approach towards making a two-level ontology out of a one-level local ontology when a global two-level reference ontology is available for a domain.

With a global ontology, the problem of generating an upper-level knowledge structure for a local ontology is reduced to the easier task of locating local concepts in the global bottom-level structure. For every local concept found in the global bottom-level structure, its semantic type may be assigned as semantic type in the newly generated upper level of the local ontology.

As a first impression, the presented problem had not appeared too difficult, given the large resources available in the UMLS and WordNet. Indeed, our initial plan was to attack semantic enrichment *without* a global ontology immediately. However, it turned out that even with the UMLS as a global ontology, the problem of semantically enriching two real, existing, small terminologies, the ACC and STS, was difficult. Even our solution for this limited problem is semi-automatic, meaning that in a few cases a human had to make the final judgment on a match.

The main source of our problems was the poor and inconsistent structure of the ACC and the STS. Because several different relationships have been used to connect concepts and categories *without distinguishing between those relationships*, categories were initially not helpful at all. Thus, we performed an extensive analysis of the different relationships that were used for connecting concepts with categories. In the end, we managed to make good use of the categories in the (surprisingly many) cases where the ACC and STS concepts did not have corresponding concepts among the nearly one million UMLS concepts. Thus we presented an algorithm for semantic enrichment that makes use of categories whenever they are available, and an architecture of how semantic enrichment is implemented as part of a larger project on ontology integration. We defined cases when semantic enrichment would create unwanted redundancies and provided experimental result data on how many such redundancies occurred for the ACC and STS (Table 8). Lastly, we showed that, with flexible matching, semantic enrichment was possible for almost all concepts, and human intervention was necessary only in a few cases (Table 6).

In future work we will primarily follow three directions:

(1) We will try to completely automate the matching process by incorporating expert knowledge that comes to bear in cases where our current algorithm still fails.
(2) We will extend our research to the case where no global ontology is available at all. Thus, semantic types need to be found from the bottom-level hierarchy itself, to create a two-level structure. This was the problem that had we wanted to solve originally.
(3) We will investigate how our solutions scale when applying them to larger terminologies. While larger terminologies will require more matching rules, we expect that the increase will be sublinear.

## Acknowledgments

This paper is based on ideas that were developed together with Yehoshua Perl during many years of research on partitioning, visualizing, and auditing medical terminologies.

## References

[1] Geller J, Gu H, Perl Y, Halper M. Semantic refinement and error correction in large terminological knowledge bases. Data and Knowledge Engineering 2003;45(1):1–32.
[2] Humphreys BL, Lindberg DA. Building the Unified Medical Language System. In: The 13th annual symposium on computer applications in medical care. Washington; DC. 1989.
[3] Humphreys BL, Lindberg DA, Schoolman HM, Barnett GO. The Unified Medical Language System: an informatics research collaboration. J Am Med Inform Assoc 1998;5(1):1–11.
[4] Lindberg DA, Humphreys BL, McCray AT. The Unified Medical Language System. Methods Inf Med 1993;32(4):281–91.
[5] Miller GA. WordNet: a lexical database for English. Commun ACM 1995;38(11):39–41.
[6] Niles I, Pease A. Linking lexicons and ontologies: mapping WordNet to the suggested upper merged ontology. In: International conference on information and knowledge engineering (IKE'03). 2003.
[7] Schuyler PL, Hole WT, Tuttle MS, Sherertz DD. The UMLS Metathesaurus: representing different views of biomedical concepts. Bull Med Libr Assoc 1993;81(2):217–22.
[8] Tuttle MS, Sherertz DD, Olson NE, Erlbaum MS, Sperzel WD, Fuller LF, Nelson SJ. Using meta-1 the first version of the UMLS Metathesaurus. In: The 14th annual SCAMC. 1990. p. 131–5.
[9] McCray AT. UMLS Semantic Network. In: The 13th annual SCAMC. 1989. p. 503–7.
[10] McCray AT. Representing biomedical knowledge in the UMLS Semantic Network. In: Broering NC, editor. High-performance medical libraries: advances in information management for the virtual era. Westport, CT: Meckler; 1993. p. 45–55.
[11] McCray AT, Hole W. Thescope and structure of the first version of the UMLS Semantic Network. In: The Fourteenth Annual SCAMC. 1990. p. 126–30.
[12] Tringali M, Hole W, Srinivasan S. Integration of a standard gastrointestinal endoscopy terminology in the UMLS Metathesaurus. Proc AMIA Symp 2002:801–5.
[13] Burgun A, Bodenreider O. Methods for exploring the semantics of the relationships between co-occurring UMLS concepts. Medinfo 2001;10(1):171–5.
[14] McCray AT, Nelson SJ. The representation of meaning in the UMLS. Methods Inf Med 1995;34(1–2):193–201.
[15] Stumme G, Maedche A. FCA-MERGE: bottom–up merging of ontologies. In: The 17th International Joint Conference on Artificial Intelligence (IJCAI). 2001.
[16] Bodenreider O, Mitchell J, McCray AT. Evaluation of the UMLS as a terminology and knowledge resource for biomedical informatics. Proc AMIA Symp 2002:61–5.
[17] Pisanelli D, Gangemi A, Steve G. An ontological analysis of the UMLS Methathesaurus. Proc AMIA Symp 1998:810–4.
[18] Lee Y, Supekar K, Geller J. Ontology integration: experience with medical terminologies. Comput Biol Med 2005 [in press].
[19] Gu H, Perl Y, Geller J, Halper M, Liu L, Cimino J. Representing the UMLS as an object-oriented database: modeling issues and advantages. J Am Med Inform Assoc 2000;7(1):66–80.
[20] Peng Y, Halper M, Perl Y, Geller J. Auditing the UMLS for redundant classifications. Proc AMIA Symp 2002:612–6.
[21] Brachman RJ, Schmolze J. An overview of the KL-ONE knowledge representation system. Cogn. Sci. 1985;9(2):171–216.
[22] Noy NF, Hafner CD. The state of the art in ontology design, in AI magazine. 1997;18(3):53–74.
[23] Zhang L, Perl Y, Halper M, Geller J, Cimino J. Enriching the structure of the UMLS Semantic Network. Proc AMIA Symp 2002:939–43.
[24] Burgun A, Hill L, Bodenreider O. Mapping the UMLS Semantic Network into general ontologies. Proc AMIA Symp 2001:81–5.
[25] D'Souza DF, Wills AC. Objects, components, and frameworks with UML: the catalysis approach. Reading, MA: Addison-Wesley; 1999.
[26] Burgun A, Bodenreider O. Aspects of the taxonomic relation in the biomedical domain. In: Proceedings of the international conference on Formal Ontology in Information Systems—Volume. Ogunquit, Maine, USA: ACM Press: 2001, p. 222–33.
[27] Pustejovsky J, Rumshisky A, Castao J. Rerendering semantic ontologies: Automatic extensions to UMLS through corpus analytics. In: REC 2002 Workshop on Ontologies and Lexical Knowledge Bases. 2002.
[28] Desmontils E, Jacquin C, Simon L. Ontology enrichment and indexing process 2003.
[29] Berlin J, Motro A. Database schema matching using machine learning with feature selection. In: The 14th international conference on advanced information systems engineering (CAiSE02). 2002.
[30] Gupta A, Ludäscher B, Martone ME. Knowledge-based integration of neuroscience data sources. In: The 12th international conference scientific and statistical database management systems. 2000. p. 39–52.
[31] Gupta A, Ludäscher B, Martone ME. Registering scientific information sources for semantic mediation. In: The 21st international conference on conceptual modeling (ER). 2002. p. 182–98.
[32] Chen L, Shadbolt NR, Tao F, Puleston C, Goble C, Cox SJ. Exploiting semantics for e-science on the semantic grid. In: Web intelligence (WI2003) workshop on knowledge grid and grid intelligence. 2003. p. 122–32.
[33] Colomb R. Formal versus material ontologies for information systems interoperatiUniversity of Queensland. St. Lucia: University of Queensland; 2004.

[34] Guarino N, Welty C. Evaluating ontological decisions with OntoClean. Commun ACM 2002;45(2):61–5.

[35] Gangemi A, Guarino N, Masolo C, Oltramari A, Schneider L. Sweetening ontologies with dolce. In: The 13th international conference on knowledge engineering and knowledge management (EKAW02). 2002.

[36] Chalupsky H. Ontomorph: a translation system for symbolic knowledge. In: Principles of knowledge representation and reasoning. Los Altos, CA: Morgan Kaufmann; 2000.

[37] McGuinness D, Fikes R, Rice J, Wilder S. The CHIMAERA ontology environment. In: The 17th national conference on artificial intelligence. 2000.

[38] Mitra P, Wiederhold G, Jannink J, Semi-automatic integration of knowledge sources. in Fusion'99. 1999.

[39] Noy NF, Musen M, Prompt: Algorithm and tool for automated ontology merging and alignment. In: The National Conference on Artificial Intelligence. 2000.

[40] Maedche AA. machine learning perspective for the Semantic Web. In: Semantic web working symposium (SWWS). 2001.

[41] Parent C, Spaccapietra S. Issues and approaches of database integration. Commun ACM 1998;41(5):166–78.

[42] Chawathe SS, Garcia-Molina H, Hammer J, Ireland K, Papakonstantinou Y, Ullman JD, Widom J. The Tsimmis project: integration of heterogeneous information sources. In: The 10th meeting of the information processing society of Japan. 1994. p. 7–18.

[43] Knoblock CA, Minton S, Ambite JL, Ashish N, Modi P, Muslea I, Philpot AG, Tejada S. Modeling web sources for information integration. In: The 15th national conference on artificial intelligence. 1998.

[44] Calvanese D, Giuseppe DG, Lenzerini M. Ontology of integration and integration of ontologies. In: The 2001 description logic workshop (DL2001). 2001.

[45] Lacher M, Groh G, Facilitating the exchange of explicit knowledge through ontology mappings. In: The 14th International FLAIRS conference. 2001.

[46] Doan A, Madhavan J, Domingos P, Halevy A. Learning to map between ontologies on the Semantic Web. In: The 11th International WWW conference. 2002.

[47] Melnik S, Molina-Garcia H, Rahm E. Similarity flooding: a versatile graph matching algorithm. In: The International Conference on Data Engineering (ICDE). 2002.

[48] Doan A, Domingos P, Halevy A. Reconciling schemas of disparate data sources: a machine-learning approach. In: SIGMOD. 2001. p. 509–520.

[49] Rodriguez MA, Egenhofer MJ. Determining semantic similarity among entity classes from different ontologies. Knowl Data Eng IEEE Trans 2003;15(2):442–56.

[50] Noy N, Musen M. Anchor-PROMPT: using non-local context for semantic matching. In: The workshop on ontologies and information sharing at the international joint conference on artificial intelligence (IJCAI). 2001.

[51] Do H, Rahm E. COMA—a system for flexible combination of schema matching approaches. In: The 28th conference on very large databases (VLDB). 2002.

[52] Madhavan J, Bernstein PA, Rahm E. Generic schema matching with CUPID. VLDB J 2001:49–58.