ELSEVIER

Computers in Biology and Medicine 36 (2006) 893–919

Computers in Biology and Medicine

www.intl.elsevierhealth.com/journals/cobm

Ontology integration: Experience with medical terminologies

Yugyung Lee[a,*], Kaustubh Supekar[b], James Geller[c,1]

[a]School of Computing and Engineering, University of Missouri, Kansas City, MO 64110, USA
[b]Stanford Medical Informatics, Stanford University, Stanford, CA 94305, USA
[c]CS Department, New Jersey Institute of Technology, Newark, NJ 07102, USA

Received 30 May 2004; accepted 7 February 2005

Abstract

To build a common controlled vocabulary is a formidable challenge in medical informatics. Due to vast scale and multiplicity in interpretation of medical data, it is natural to face overlapping terminologies in the process of practicing medical informatics [A. Rector, Clinical terminology: why is it so hard? Methods Inf. Med. 38 (1999) 239–252]. A major concern lies in the integration of seemingly overlapping terminologies in the medical domain and this issue has not been well addressed. In this paper, we describe a novel approach for medical ontology integration that relies on the theory of Algorithmic Semantic Refinement we previously developed. Our approach simplifies the task of matching pairs of corresponding concepts derived from a pair of ontologies, which is vital to terminology mapping. A formal theory and algorithm for our approach have been devised and the application of this method to two medical terminologies has been developed. The result of our work is an integrated medical terminology and a methodology and implementation ready to use for other ontology integration tasks.
© 2005 Elsevier Ltd. All rights reserved.

Keywords: Terminology; Ontology; Ontology integration; Semantics; Semantic enrichment; Semantic refinement; Semantic integration

1. Introduction

In medical research, there is a need to exchange valuable information between different researchers or research groups, for the purpose of independent analysis or the verification of experimental results.

* Corresponding author. Tel.: +1 816 235 5932; fax: +1 816 235 5159.
E-mail address: leeyu@umkc.edu (Y. Lee).

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

Increasingly, we are also seeing the emergence of distributed scientific processing. The Internet provides an important platform for this activity of medical information exchange to take place. However, there are still some difficulties to resolve before seamless interoperability and interchange can occur. The main cause for these limitations arise from the fact that different research groups rely on heterogeneous research data sources.

Over the past several years, one field that has been very active in building and using terminologies, ontologies and vocabularies has been medical informatics [1]. There exist a large number of terminologies developed for different purposes (literature indexing and retrieval, electronic patient records, statistical reports on mortality, billing), in different subdomains (diseases, genomes, micro-organisms, diagnoses, medical devices, procedures, drugs). These terminologies have been built by different institutions (World Health Organization, National Library of Medicine, College of American Pathologists, etc.) on different continents, for different purposes. Yet, attempts to represent the whole medical domain are usually limited in scope (GALEN) [2] or lack a strong organizational structure, as in the Unified Medical Language System (UMLS). The UMLS [3] is a compendium of about 100 individual terminologies, but integrating a new terminology into it is a difficult task. Existing methods for integration of ontologies use structural and semantic methods (see Section 6), however, there is still room for improvement.

Most ontologies [4] are organized around a concept hierarchy (a tree or a directed acyclic graph). Many ontologies add rules, axioms, or other additional mechanisms to this backbone. In this paper, as a first step, we will only deal with integration of the concepts in the concept hierarchy. When integrating two ontologies, it is necessary to identify pairs of concepts that have the same meaning in both of them. Clearly, these concepts should occur only once in an integrated ontology. The existence of synonyms and homonyms causes problems for this kind of integration. However, a much bigger problem is the existence of subtle differences between *implemented* concepts that have the same name and stand, vaguely, for the same (concrete or abstract) real world objects. These differences occur because different ontology designers may bring different world views to the task, conceptualizing the world at different levels of granularity and abstraction. Such differences are commonly considered semantic problems.

Implementations add structural problems also. We note that for human communication the lack of exact matches does not normally make communication impossible. For the example of a calendar date, people might greatly disagree on the exact structure *beyond the obvious data type and name issues*. Thus, a person A may use the US calendar. However, A is a professor, and his academic calendar defines the Wednesday before Thanksgiving to be a Friday. Thus, A distinguishes between two kinds of dates. For A, any date needs to be annotated as a US date or an academic date. Person B, on the other hand, might not know about academic calendars and will not need any additional annotations. Clearly, A and B have concepts of calendar dates that are sufficiently different to make an exact match impossible. Nevertheless, A and B are able to communicate about dates in most situations. Thus, differences between concept representations should not automatically exclude matches.

In this paper, we propose to address some issues in ontology integration of medical terminologies by extending our semantic refinement methodology, called algorithmic semantic refinement (ALSER) [5,6]. To capture the essence of ALSER in two sentences, we "compute" small sets of concepts of similar semantics based on a superficial human specification that does *not* have to be free of contradictions. The "computation" is based on intersections of concept sets. Our methodology was previously applied to the problem of auditing the very large UMLS [6].

The ALSER methodology is based on a kind of terminology that we call a *terminological knowledge base* (TKB). However, there are many important medical ontologies or controlled medical terminologies, which are not structured as TKB. Thus, we first address the question of how to transform existing medical terminologies into a TKB. Due to the great difficulty of this problem, we are only addressing the case where a global reference ontology (i.e., UMLS) exists in the same domain as the given medical terminology. Our methodology, called *semantic enrichment*, builds a TKB by finding its concepts in a global ontology that has semantic types. Secondly, we describe the ALSER methodology generating precomputed sets of semantically similar terms in both ontologies that need to be integrated. Finally, we introduce the *semantic integration* methodology, called SEMINT, which compares and integrates terms from two ontologies, if they are already classified as semantically similar.

We believe that this *incremental* approach composed of *semantic enrichment*, *semantic refinement* and *semantic integration* will reduce the likelihood of false positives, since we avoid matching concepts of different semantics, which out of principle cannot be the same. The semantic approach has an additional advantage. It greatly reduces the computational effort of the matching operations. Fewer terms will have to be matched against each other.

In Section 2, we describe the difficulties in handling medical terminologies and also propose our semantic enrichment approach. In Section 3, we introduce our methodology of ALSER. In Section 4, we show how this methodology can be used to create pairs of candidate sets of concepts that have to be matched with each other. A prototype implementation and results are described in Section 5. In Section 6, we review related research work. Section 7 contains our conclusions.

2. Semantic enrichment

The semantic enrichment process of finding semantic types for concepts is difficult, even in the medical domain, with the UMLS readily available. In this section, we briefly survey the two medical terminologies to describe some obstacles that we have encountered during the integration of medical ontologies and to highlight the necessity of semantic enrichment as a precursor of ontology integration. The American College of Cardiology (ACC) has provided a list of 142 terms with definitions [7]. These concepts are separated into 22 "categories." The Society of Thoracic Surgery (STS) has created a classification of 248 terms, subdivided into 21 categories [8].

2.1. The Unified Medical Language System

The UMLS [9–11,3], designed by the National Library of Medicine (NLM), consists of three knowledge sources of which we are interested in two, the Metathesaurus [12,13] and the Semantic Network [14–16]. The *Metathesaurus* is a unified collection of many different medical terminologies. It is a compilation of terms, concepts, relationships, and associated information. The January 2003AA edition includes 875,255 concepts and 2.14 million concept names in over 100 biomedical source vocabularies.[2]

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

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

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 $\geqslant$ 24Hrs	Complications

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.

2.2. *Difficulties in handling medical terminologies*

Both the above ontologies, ACC and STS, 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, categories should have been introduced for the purpose of categorizing concepts (similar to semantic types in the UMLS). This is what the name "categories" seems to imply. However, as mentioned before, there exist different kinds of relationship between concepts and categories of the ACC and STS. 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, we use the category information provided, according to our understanding of the relationship that is presumably holding between the concept and the category.

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 terminologies, 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. In order to handle acronyms, a

Table 2
Complications that appeared in ACC/STS concept names

Pattern	Name	Description
Instance-of	Valve disease—Mitral	*Mitral* is an instance of *Valve Disease.*
Acronym-of	VS-Aortic Proc-Procedure	*VS* is a *Valve Surgery.*
	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.
Redundant word	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	$\geqslant$ and "*Greater than Equal*"	Different notations for the same concept

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* (see Section 2.2).

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* $\geqslant$ 24 Hrs 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., Mitral 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.

2.3. Details of data enrichment

By first classifying the concepts of a new terminology using the Semantic Network of the UMLS, this task becomes more manageable. A new concept does not have to be compared with every concept in the UMLS, 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.

In order 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 instance, 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. We call this expansion step *data enrichment*. Data enrichment also performs other preprocessing and clean-up steps, such as dealing with non-alphabetic characters occurring in many medical terms.

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 Date of Birth (DOB), Myocardial Infarction (MI), and many more. Whenever terms contain special characters such as "/" or "-" they are replaced by blanks. When there exist prefix or postfix cases (e.g., RF-Smoker), they are converted into a special form (e.g., Risk Factor:Smoker). In this way, those terms can be matched with other terms of the same meaning which are lacking the special characters. In some cases, precise mathematical symbols are expressed by imprecise English words. For example, the mathematical notation "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, 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. The synonyms of terms were derived both from STS and ACC documentation and from the UMLS (currently still manually, in a preprocessing step).

2.4. Details of semantic enrichment

We now explain the semantic enrichment process in more detail (Fig. 1). Semantic types are drawn as little squares and named with capital letters, while concepts are drawn as circles and named with small letters. ACC and STS categories are drawn as filled in circles. The medical terminologies (ACC, STS) do not have semantic types. Thus, a global ontology (the UMLS) is used to add semantic-type assignments for the concepts in the ACC and the STS.

There are two possible cases how semantic enrichment can be done. In Case 1, a concept of the local terminologies is found in the global ontology. This concept has one or several assigned (global) semantic types which are returned. In Fig. 2, the concept a matches the global concept z. This might happen if a is a synonym of z in the global ontology. Because z has assigned semantic types U and S, a acquires U and S as its semantic types.

In Case 2, a local concept without a match in the global terminology belongs to a category such that an IS-A relationship holds between the concept and the category. In this case it is reasonable to look in the UMLS for the semantic type of the category and assign this semantic type to the concept for the following reason. In some cases, concepts are ambiguous. However, the category eliminates this ambiguity. Thus, the semantic type of the category should help to better define the meaning of the concept.

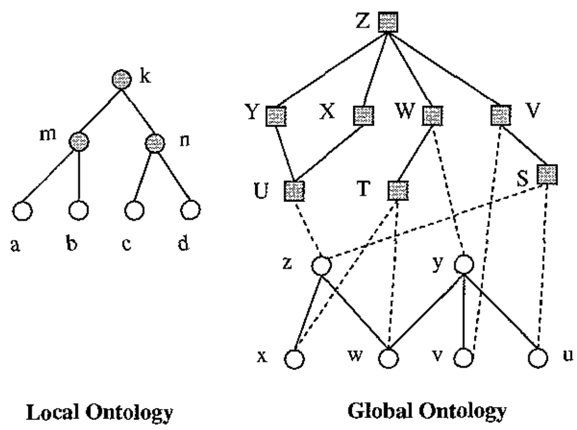

Fig. 1. Local and global ontologies.

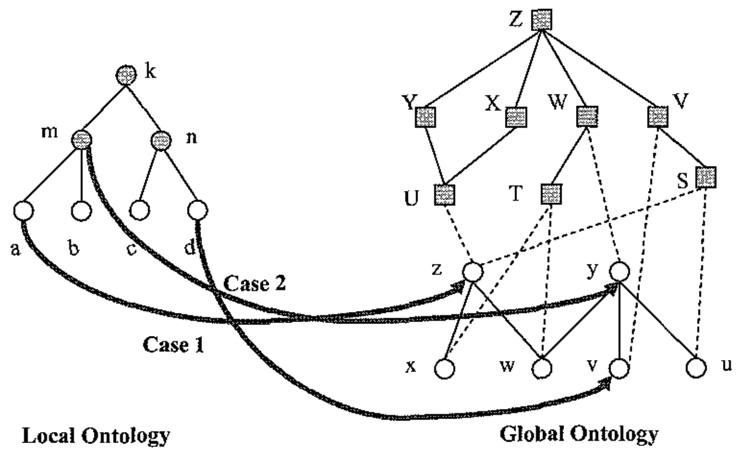

Fig. 2. Mapping between local and global ontologies.

In Fig. 1, the concept b is connected to the category m by an IS-A link. The category m matches the concept y, which has a semantic type W assigned to it. Thus, we assign the semantic type W to the concept b. But a is also connected to the category m by an IS-A link. Thus, a now gets the additional semantic type W assigned to it, resulting in a having S, U, and W as semantic types (see Fig. 3).

We note that whenever a concept is assigned several semantic types, we make sure that there are no *redundant* assignments. The assignment of a concept to a semantic type R_p is *redundant* if that concept is also assigned to another semantic type R_c, and R_p is a parent or ancestor of R_c in the semantic network [6,17]. For example, in Fig. 2, assigning the semantic types W and Z to the concept w would be redundant, because w is already assigned to T, and W is a parent of T and an Z is an ancestor of T.

Fig. 3 shows the results of semantic enrichment. The concepts c and d are crossed out, because there were no semantic types corresponding to them in the global ontology. We do not allow concepts without

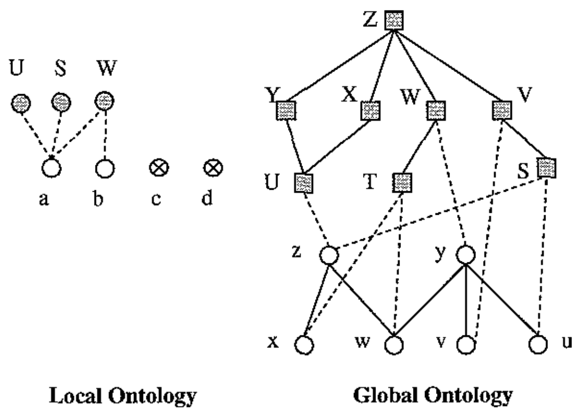

Fig. 3. Enrichment of local terminology after the mapping.

semantic types. In our experiments, these cases were rare. Only 4 out of 248 concepts were dropped for this reason.

3. Semantic refinement

3.1. Methodology for semantic refinement

ALSER was invented [6,5] in the context of the UMLS. We formally present the ALSER methodology that we have published in [5].

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 TKB. Thus, a TKB is a triple:

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

in which $\hat{\mathscr{C}}$ is a set of concepts, $\hat{\mathscr{S}}$ is a set of semantic types, and μ is a set of assignments of concepts to semantic types. We will use capital letters for semantic types and small letters for concepts.[3] Finally, μ consists of pairs (c, S) such that the concept c is assigned to the semantic type S:

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

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

In [5], $\hat{\mathscr{S}}$ and $\hat{\mathscr{C}}$ form DAG structures. In this paper, they are sets. Furthermore, it holds:

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

In words, every concept must be assigned to at least one semantic type. The opposite condition does not hold. ALSER takes as input one TKB and returns as result a second TKB with better semantics. What

[3] Both roman and italic fonts.

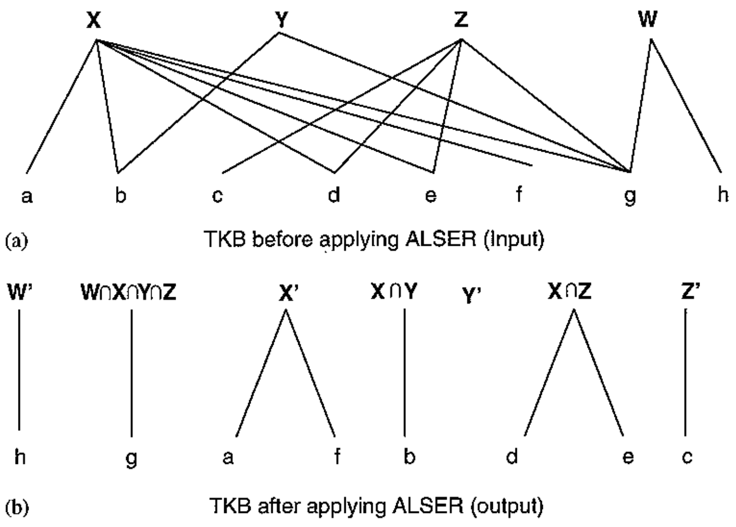
X
Y
Z
W
a
b
c
d
e
f
g
h
(a)
TKB before applying ALSER (Input)
W'
W∩X∩Y∩Z
X'
X∩Y
Y'
X∩Z
Z'
h
g
a
f
b
d
e
c
(b)
TKB after applying ALSER (output)

Fig. 4. Algorithmic semantic refinement example in graph notation.

"better semantics" exactly means will become clear by the end of this section. The refinement can be written as a function

$$\text{TKB}' = \text{ALSER}(\text{TKB}), \quad \text{TKB} = \langle \hat{\mathscr{C}}, \hat{\mathscr{S}}, \mu \rangle. \tag{5}$$

Next, we need a notion when two concepts are semantically similar. We will consider two concepts c, d as similar, $c \simeq d$, when they are assigned to exactly the same set of semantic types of a TKB:

$$c \simeq d: \quad \forall S \in \hat{\mathscr{S}}[(c, S) \in \mu \Leftrightarrow (d, S) \in \mu]. \tag{6}$$

Intuitively, if two concepts c, d are assigned to exactly the same semantic type X, then we can say that these two concepts have similar semantics. If two concepts c, d are assigned to X and are also assigned to Y, then these two concepts also have similar semantics. On the other hand, if a concept a is assigned to X and a concept b is assigned to both X and Y, then a and b will have semantics that are not similar in the formal sense defined above. (They are still similar, to a lower degree, in the real world.)

We will use a running example to explain our rather abstract definitions. Fig. 4(a) shows a TKB before applying the ALSER algorithm. Fig. 4(b) shows the corresponding TKB$'$ that is the output of ALSER. Capital letters represent semantic types, small letters represent concepts. A line connecting a semantic type to a concept means that this concept is assigned to this semantic type. The following is an informal explanation of Fig. 4, which will be formalized shortly. As the figure shows, the result of ALSER is that semantic types are refined, and every concept in the output is assigned to exactly one refined semantic type. Thus, refined semantic types never share concepts with one another. All concepts assigned to any one refined semantic type in the output are similar ($\simeq$).

Definition 2 (*Original semantic type*). A semantic type of a given TKB (i.e., one that is intended as input to ALSER) is called an original semantic type. *Example*: X, Y, Z and W are original semantic types in Fig. 4(a).

Definition 3 (*Simple concept*). A simple concept is assigned to one single semantic type. *Example*: a, c, f and h are simple concepts in the input.

Definition 4 (*Compound concept*). A compound concept is assigned to two or more semantic types. *Example*: b, d, e, and g are compound concepts in the input in Fig. 4(a).

Definition 5 (*Refined semantic type*). Every semantic type which is part of the output of ALSER is called a refined semantic type. *Example*: X', Y', Z', W', $X \cap Y$, $X \cap Z$, $W \cap X \cap Y \cap Z$ are refined semantic types in Fig. 4(b).

Definition 6 (*Simple semantic type*). A simple semantic type P' is a refined semantic type defined to be assigned all and only the simple concepts assigned to the original semantic type P. *Example*: The refined semantic types X', Y', Z' and W' are simple semantic types.

Definition 7 (*Intersection type*). An intersection type I is a refined semantic type defined to be assigned all and only the compound concepts assigned to exactly one specific set of original semantic types. An intersection type is denoted by the sequence of those original semantic types, where the intersection symbol appears between any two consecutive original semantic types in the sequence. *Example*: $X \cap Y$, $X \cap Z$ and $W \cap X \cap Y \cap Z$ are intersection types.

The intersection type $W \cap X \cap Y \cap Z$ is the same as the intersection type $Y \cap X \cap W \cap Z$, etc. For clarity, an intersection type may have a shorter direct name which may be chosen to reflect the semantics of the concepts. Many times there are obvious names for intersection types. For example, the intersection type of the semantic type body part with the semantic-type manufactured object is commonly known as prosthesis.

Definition 8 (*Semantically uniform TKB*). A TKB is called semantically uniform if and only if each concept is assigned to exactly one semantic type.

In summary, the input to ALSER consists of concepts which may be assigned to several semantic types, which is undesirable.[4] In the output of ALSER, every concept is assigned to a single refined semantic type. Therefore, all concepts assigned to any one refined semantic type are now similar, according to the previous definition of similarity $\simeq$. The output of ALSER may contain empty semantic types.

As even our small example shows, the output will (normally) contain more (refined) semantic types than the input had (original) semantic types. Each simple semantic type has, on average, fewer concepts assigned than the corresponding original semantic type. We will show this effect with real data in Section 5. We can now advance to a procedural definition of ALSER. The easiest part of the ALSER algorithm deals with the set of concepts. ALSER does not change the number or names of concepts in a TKB. It only reassigns the concepts to refined semantic types

$$\hat{\mathscr{C}}' = \hat{\mathscr{C}}. \tag{7}$$

[4] Why it is undesirable was explained in great detail in [6,5]. From the integration point of view, it is impossible to tell whether two compound concepts are similar without looking at all semantic types they are assigned to.

In order to find the set $\hat{\mathscr{S}}'$, we first need to identify the simple semantic types. As noted above, every original semantic type with at least one simple concept is mapped into a simple semantic type. All the simple concepts assigned to an original semantic type are reassigned in the output to the corresponding simple semantic type. In the second step, we need to identify all intersection types. Every compound concept identifies a set of original semantic types. Every such set needs to be mapped into one intersection type. The compound concept will then be reassigned to this intersection type in the output. Formally, for every combination of k original semantic types $S_{i_1}, S_{i_2}, S_{i_j}, \ldots, S_{i_k}$ from TKB such that there is at least one single concept c that is assigned to all S_{i_j}, for $1 \leqslant j \leqslant k$, we create an intersection type $S_{i_1} \cap S_{i_2} \cap \cdots \cap S_{i_k}$ in TKB$'$.

The derivation of μ' also consists of two parts. μ'_s contains the assignments of concepts to simple semantic types. μ'_i contains the assignments of concepts to intersection types. For a specific simple semantic type X', μ'_s can be found as follows:

$$\mu'_s = \{(c, X') | \text{simple}(c) \& (c, X) \in \mu\}. \tag{8}$$

4. Semantic integration

In the real world, there is a spectrum of requirements one could impose to accept two concepts as matching. On one extreme, one might insist that there be only perfect matches between two concepts. In the example of calendar dates, one might require that both are called "date" and have exactly three fields, two numbers and one character string (for the month). Furthermore, there also must be a match between the attribute names. For example, one might insist that both of them have an attribute called "month." Such requirements would lead to a few matched concepts and large numbers of unmatched concepts.

The other extreme is to insist that all (or almost all) concepts of the smaller ontology are matched against concepts in the larger ontology, as long as there is at least some structural similarity. This extreme could be based on the assumption that both ontology designers did a reasonable job to cover the domain, and thus a fundamental concept such as "date" simply has to appear in both ontologies, no matter what it is called, and no matter how exactly it is structured. Our solution is closer to the second extreme.

4.1. Which pairs need to be matched?

We now assume that we have to integrate two TKB and TKB$_2$. Before we can integrate, we need to identify for which concepts to attempt integration. This requires three steps: (1) We transform TKB into TKB$'$ by applying ALSER. (2) We transform TKB$_2$ into TKB$'_2$ by applying ALSER. Step (2) is the transition that happens from the right top part of Fig. 5 to the right middle part of the figure. Note that concepts of TKB are assigned by semantic types of UMLS during semantic enrichment. Thus, *Cardiogenic Shock* and *Arrhythmia Type* are assigned to *Functional Concept* and *Therapeutic or Preventive*, *Diabetes* is assigned to *Disease or Symptom*, etc. The set of refined semantic types of TKB$'_2$ is not necessarily the same as the set of semantic types of TKB$'$. In our figure, TKB$'_2$ has an additional intersection type *Therapeutic or Preventive* $\cap$ *Disease or Symptom* which does not occur in TKB$'$. It is assigned the concept *Congestive Heart Failure*. (3) We identify pairs of sets in TKB$'$ and TKB$'_2$ that are potential matches (Fig. 5). At the bottom of Fig. 5, no match will be attempted for the concept *Congestive Heart Failure*. An

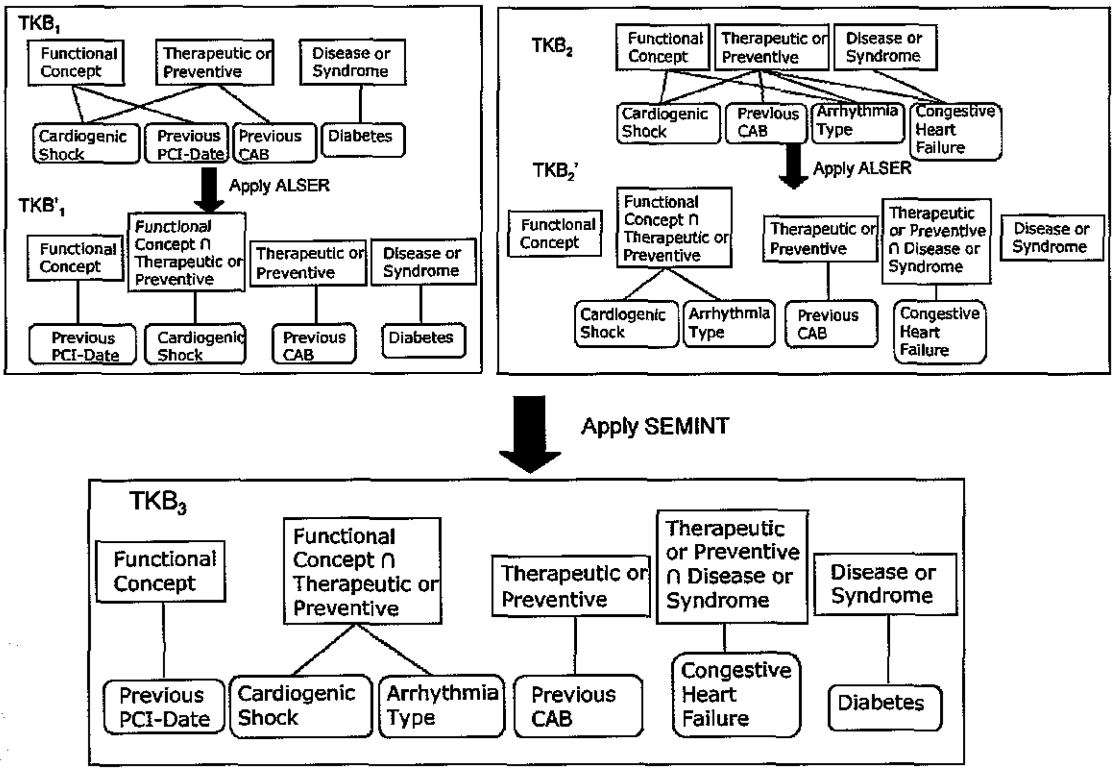

Fig. 5. Overall process of semantic integration.

attempt is made to match the concepts in the set {*Cardiogenic Shock, Arrhythmia Type*} against the concept *Cardiogenic Shock*. Similarly, the match of *Previous CAB* and *Previous CAB* is attempted. Formally:

1. Assume a refined semantic type S exists in TKB$'$ that has assigned concepts $x, y, z, \ldots$. Further, assume that S does not exist in TKB$'_2$ or, there are no concepts assigned to S in TKB$'_2$. Then, by the similarity assumptions made above, no concepts corresponding to x, y, $z, \ldots$ exist anywhere inTKB$'_2$. Thus, these concepts do not need to be matched at all. Alternatively, the concepts $x, y, z, \ldots$ might have been misclassified along the way [5] by human experts. In that case, the integration effort would make those problematic concepts easier to detect [5].
2. The above observation applies in reverse also. If an intersection type S has been generated for TKB$'_2$ that does not exist in TKB$'$, then the concepts x, y, $z, \ldots$ assigned to S will not have corresponding concepts anywhere in TKB$'$. Thus, these concepts do not need to be matched at all. The same observation as above applies to possible misclassifications.
3. All pairs of concepts (q, r) with $q \in$ CONCEPTS(S, TKB$'$) and $r \in$ CONCEPTS(S, TKB$'_2$) are similar ($\simeq$) and need to be matched. The matching algorithm follows in Section 4.2.

A brute force approach would be to compare every concept from TKB$'$ with every concept from TKB$'_2$. In our approach, we only compare pairs of concepts for which we already know that they are similar.

4.2. Scoring concept similarities

Now, we describe details of how scores for concept similarities are computed. We use three aspects to determine whether a match exists between (1) concepts, (2) attributes of concepts, (3) relationships that point from one concept to another concept.

4.2.1. Ranking concepts by terms

If two concepts have similar names then they are possibly matches. We used the bigram approach that is known to be a very effective, simply programmed means of determining a similarity measure between strings. The bigram approach consists of three steps. (1) Sequences of two consecutive letters within a string are extracted (e.g., the word "heart" contains 'he', 'ea', 'ar' and 'rt'); (2) Two sequences of bigrams are compared, and a raw similarity score is computed; (3) A matched score is computed from the raw score, i.e., the number of the common bigrams is divided by the average number of bigrams in the two strings.

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). We 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.

4.2.2. Ranking candidates by attributes

Assume that we are given a pair of concepts from two different ontologies. These concepts have different terms, therefore, a priori there is no reason for a computer to assume that they are in fact describing the same concepts. In order to establish whether they are indeed the same concept, we need to compare attributes.

We assign to every pair of concepts a score as follows.

1. Two concepts that have the same number of attributes, and for every attribute in one concept there is an attribute in the other concept of the same name and same data type, are considered perfectly matched, with a score of 1.
2. If two attributes (of two concepts from two ontologies) have the same name but are of different data types, we assign them a score of $k(k < 1, k \geqslant 0)$.
3. Then we compute the ratio of matched attribute scores divided by the number of attributes of the concept that has more attributes.
4. The final decision about similarity is made, based on a minimum threshold for the computed combined

4.2.3. Ranking candidates by relationships using propagation

In the previous steps, we have established matches between concepts from two different ontologies, based on pairs of terms and attributes. However, two concepts that point to exactly the same concepts with the same relationships are presumably very similar to each other. We view the relationship targets as data types, and two concepts that point to all the same data types are likely to be quite similar. However, we would have a chicken and egg problem here, if we start with considering relationships from the beginning. That is, the case because the relationships targets cannot be used for matching if they themselves have not been matched up.

This is why we start by matching up a few concepts using terms and attributes alone. By this step, we create an initialization for matching up additional concepts by using relationships. Thus, two concepts with different names that point to several target concepts that all have been matched up between two ontologies are presumably themselves a match. We can use a similar ratio criterion as for attributes, however, now the targets carry more semantics than the undifferentiated data types of attributes. Thus, we are willing to assign a pair of relationships a high score if the targets are the same OR if the relationship names are the same. Let us assume now that a set of concept pairs has been established such that the concepts in each pair match and are from two different ontologies. Then any pair of concepts that point to these matched concepts would also be considered highly ranked for being matches. Thus, after establishing initial matches, we continue ranking concepts by similarity using a process similar to a Waltz filtering [18].

Thus, the process of finding matches needs to be recomputed until a score change of one concept pair does not result in a score change of any concepts pointing to that pair anymore. Note that this state of equilibrium can be easily achieved, as we are using a threshold. If there are only changes that do not cross the threshold, the update process would terminate.

4.2.4. Combining matching scores

Two concepts are considered matched if their terms, their attributes and their relationships are (on average) similar. A weight is assigned to each similarity aspect of a concept (term similarity, average attribute similarity, average relationship similarity). Considering these three criteria, we now compute the degree of the similarity of concepts from two distinct ontologies. For this purpose, we use a multiple attribute decision making (MADM) approach, a simple additive weight-based solution [19]. This approach determines a combined score of concept matches between ontologies. Let $C_i = \{C_{i1}, C_{i2}, \ldots, C_{im}\}$ and $C_j = \{C_{j1}, C_{j2}, \ldots, C_{jn}\}$ be sets of concepts for given ontologies, and let $F = \{F_1, F_2, \ldots, F_p\}$ be a set of p features (in this paper $p = 3$) that characterize the degree of similarity. The weight vector W reflects the importance of each attribute $W = \{W_1, W_2, \ldots, W_p\}$, where $\sum W_i = 1$. We compute the scores for each of the p features for each of l matching cases ($l \gg n$ or m) in a decision matrix $D = d_{ij}$.

The method is comprised of three steps: first, scale the scores into a range [0, 1], with the best score represented by 1, using

$$r_{ij} = (d_{ij} - d_{j\,\min})/(d_{j\,\max} - d_{j\,\min}). \tag{9}$$

Second, apply weights and third, sum up the values for each of the alternatives, using

$$S_i = \sum W_j r_{ij}. \tag{10}$$

After a combined score has been computed, we compare the weighted sum with a given threshold α. Some matches may be lacking attributes or relationships. In this case, a weight of zero will be assigned to these aspects of a concept. All combined similarity values greater than α are stored in a matrix G_T. Subsequently, concept pairs with similarity values above the threshold are constructed, starting with the maximal similarity value. If there are several equal maximal similarity values, they are processed in random order. Whenever the next largest similarity value has been identified between two concepts c and d, then the complete row of c and the complete column of d in the similarity matrix G_T are set to 0. This is because c and d are not available for matching anymore.

4.2.5. *The SEMINT algorithm*

Thus, the algorithm for matching concepts from TKB$'$ with concepts from TKB$'_2$, assigned to the same semantic types, follows below. The three output lists L_1, L_2, and L_3 correspond to pairs of matched concepts (L_1) and unmatched concepts of TKB$'$ and TKB$'_2$ (L_2 and L_3, respectively). The algorithm makes use of the matrix G_T that is dynamically resized at every iteration of the main loop.

```
Algorithm. SEMINT
(Input: TKB′, TKB′_2; Output: $L_1, L_2, L_3$)
$L_1 = L_2 = L_3 = \{\}$;
FOR ALL semantic types $S \in \hat{\mathscr{S}}'_2${
    //Note that $\hat{\mathscr{S}}'$ (SemanticTypeSet(TKB′)) = $\hat{\mathscr{S}}'_2$ (SemanticTypeSet(TKB′_2))
    $G_T$[, ]; // Two dimensional array of floats for similarity values;
            //Cardinality of CONCEPT($S$, TKB′)defines number of rows.
            // Cardinality of CONCEPT($S$, TKB′_2) defines number of columns.
    FOR ALL pairs of concepts $(c, d)$,
        $c \in$ CONCEPT($S$, TKB′),
        $d \in$ CONCEPT($S$, TKB′_2 {
        $B$ = Bigram similarity of the name of $c$ and the name of $d$;
        $A$ = Attribute similarity of $c$ and of $d$;
        $R$ = Relationship similarity of $c$ and of $d$;
        // Combine these three similarities using weighted average for given weights $W_i, W_j, W_k$.
        $G = MADM(F, W)$ where $F = B, A, R$ and $W = W_i, W_j, W_k$
        IF ($G > \alpha$){//$\alpha$ is a threshold value
            $G_T[c, d] = G$;
        } ELSE {
            $G_T[c, d] = 0$;
        }
    }
    // Loop as long as there is a value greater than 0 ANYWHERE in the 2-D array $G_T$.
    // The WHILE loop hides a 2-D search.
    WHILE there are values greater than 0 in $G_T$ [,] {
        // The line below also involves a 2-D search.
        Find the maximum value in $G_T$[, ];
        Say, this maximum value is at $(c, d)$
        Store $(c, d)$ as a match in the list $L_1$;
        Set all $G_T[c, ] = 0$ // Remove $c$ from consideration
        Set all $G_T[, d] = 0$ // Remove $d$ from consideration
    }
    Append all remaining elements of CONCEPT($S$, TKB′) to the list $L_2$;
    Append all remaining elements of CONCEPT($S$, TKB′_2) to the list $L_3$;
}
```

The above algorithm performs a matching operation on pairs of concepts. The structure of the algorithm is such that two concepts are considered matched if their names, their attribute names and their relationship names are (on average) similar. Inside of the algorithm, attribute similarity and relationship similarity are

Table 3
Combining matching scores

G_T	d	e
Stage 1		
a	0.8	0.9
b	0.7	0.75
c	0.95	0.85
Stage 2		
a	0	0.9
b	0	0.75
c	0	0
Stage 3		
a	0	0
b	0	0
c	0	0

computed by subalgorithms (not shown) that perform matching operations on pairs of attribute names and relationship names, respectively. The attribute similarity is the average of all the bigram similarities of all the matched attribute names. Attributes that do not match are ignored. We can use exactly the same way to compute the relationship name similarity for given pairs of concepts. Thus, the two subalgorithms are structurally similar to the algorithm SEMINT itself. A weight is assigned to each similarity aspect of a concept (name, attribute, and relationship similarity). Then we compare the weighted sum G with a given threshold α. For ontologies that lack attributes or relationships, a weight of zero is assigned to those aspects of similarity.

To demonstrate the functioning of the above algorithm, assume that $\text{CONCEPT}(S, \text{TKB}') = \{a, b, c\}$ and $\text{CONCEPT}(S, \text{TKB}'_2) = \{d, e\}$ (Table 3). In Stage 1 G_T contains all values that are above α. First, we append (c, d) to L_1, giving $L_1 = \{(c, d)\}$. Then, we set the row of c and the column of d to 0 (Stage 2). Next, we create the pair (a, e) and append it to L_1, giving $L_1 = \{(c, d), (a, e)\}$. Now, the row of a and the column of e have to be set to zero (Stage 3). At this point, all values in G_T are zero. The concept c was not matched with any other concept and is appended to L_2. $L_2 = \{c\}$. $L_3 = \{\}$ stays empty for this semantic type. The algorithm would now advance to the next semantic type.

At this point, we have reached the limit of what can be done algorithmically. A human expert will need to review all matches in L_1, or only those with a relatively small G value (greater than α). Similarly, he will need to review L_2 and L_3 to look for any missed matches or misclassifications. By our matching philosophy, we are trying to keep L_2 and L_3 small, limiting the manual effort that goes into this step.

5. Implementation and experimental results

5.1. Implementation architecture

We have implemented an ALSER/SEMINT prototype system [20] following the paradigm of component-oriented development [21]. The component-based development approach allows a complex system

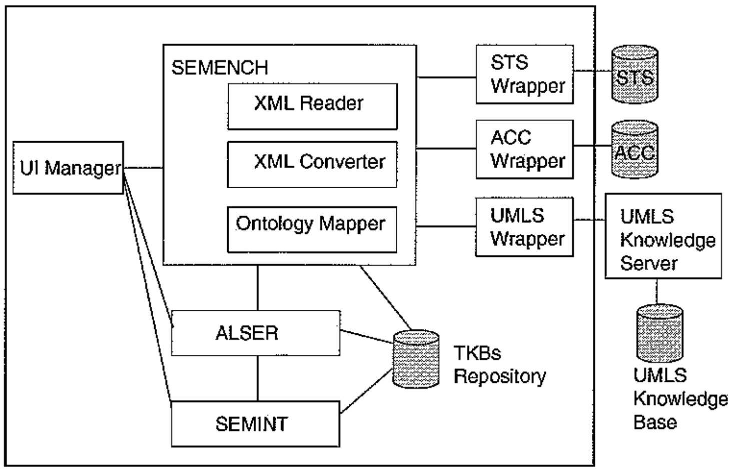

Fig. 6. The architecture.

to be considered as a composition of an arbitrary number of smaller components with well-defined interfaces. Our system architecture is shown in Fig. 6. The UI manager handles a user's requests by invoking an appropriate subsequent component. The two main components of the system are the ALSER component and the SEMINT component which have been extensively described above.

ALSER and SEMINT expect as input TKBs coded in XML. We are using XML, because it allows us to quickly extract data and exchange information between components. Unfortunately, most existing terminologies are not in that format. The *SEMENCH* component, composed of three subcomponents (XML Reader, XML Converter, and Ontology Mapper), performs the required translation of the input. If the input format is not already XML, then the ontological input data has to be transformed into XML, using the *XML Converter*. 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 [22].

Secondly, the given terminology or ontology has to be transformed into a TKB. The only preexisting knowledge base that strictly follows the TKB format is the UMLS. We performed experiments the ACC terminology and the STS terminology. Both those medical terminologies are not structured like TKBs. Thus, they first had to be transformed into TKBs. Before discussing the *Ontology Mapper* we briefly describe those two ontologies.

The *Ontology Mapper* transforms terminologies into TKBs using wrappers. In our case, three wrappers are needed, the *ACC Wrapper*, the *STS Wrapper*, and the *UMLS Wrapper*. As we will explain in the next subsection, the ontology Mapper performs the "semantic enrichment" of the two local terminologies (ACC and STS) through a global ontology, the UMLS. 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 [3]. It takes concepts as input and returns corresponding UMLS semantic types. We have found that implementing the UMLS Wrapper is difficult, as it requires Natural Language Processing. Thus, the results of the UMLS Wrapper needed to be hand-checked by a human. If the UMLS Wrapper did not find any semantic type, then the human expert found one by looking up the concept on the UMLSKS server. Secondly, if the human expert judged that

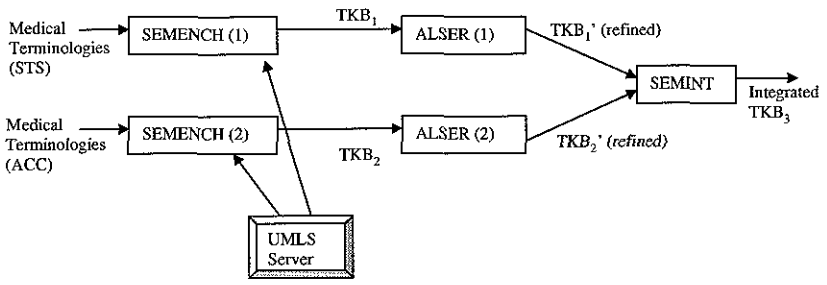

Fig. 7. The data flow of the ALSER/SEMINT framework.

Table 4
Some examples of actual mapping before/after data enrichment

Case	STS concept	ACC concept	After data enrichment
Synonym	*Date of Birth*	Patient *DOB*	Date of Birth
	Readmission Reason	*Readmit* Reason	Readmission Reason
Acronym	*RF*-Diabetes	Diabetes	Risk Factor:Diabetes
Redundant word	MI	*Previous* MI	Myocardial Infarction
	Patient ID	*Unique* Patient ID	Patient ID
	Payor	Insurance Payor	Payor
Compound word	Comps-Op-ReOp Bleed/Tamponade	Tamponade	Tamponade

the semantic type was incorrect, she looked it up on the UMLSKS server. If the semantic type appeared correct, it was left alone. Thus, in this paper, the task of the UMLS Wrapper is augmented by a human. We have collected observations how the human performs this task and intend to improve the UMLS Wrapper in future research.

The UMLSKS server offers several options. We are using "advanced search" with approximate matching. Terms are derived from all source vocabularies in the UMLS 2003AA. Using semantic enrichment, two TKBs are generated, encapsulating the ACC and STS terminologies, respectively. The TKBs generated by SEMENCH, refined by ALSER and/or integrated by SEMINT are stored in the *TKB Repository* for future use. The ALSER component needs to be invoked for both TKBs. SEMINT is then applied to the two refined TKBs, resulting from the application of ALSER (Fig. 7).

5.2. *Experimental results*

In order to test our algorithms, we have performed extensive experimental work. As mentioned above, we used the UMLS for enriching the terminologies of the ACC [7] and the STS [8]. While the goals and sources of the STS and ACC are different, there is some degree of overlap. Table 4 demonstrates the necessity of data enrichment. It shows matches of concepts in the ACC and the STS which became evident only after applying data enrichment to the terms in the table. Table 5 shows part of the STS terminology before enrichment and the enriched STS terminologies. The symbol $\bigcap$ in Table 6 indicates that a concept belongs to all the semantic types connected by $\bigcap$, i.e., the intersection type. Table 7 shows how concepts

Table 5
Partial STS terminologies before/after semantic enrichment

Category name	Concept name
Post operative Hospitalization	Initial ICU hours
	Additional ICU hours
	Total hours ICU
History and risk factors	Weight, height
	Risk Factor:Smoker,
	Risk Factor:Smoker current
Operative	Urgent reason
	Emergent reason
Diagnostic cath procedure—findings	Valve Disease:Insufficient Aortic,
	Valve Disease:Insufficient Mitral
Operative	Valve Surgery:Aortic Procedure,
	Valve Surgery:Mitral Procedure
Semantic type	**Concept name**
Temporal concept	Initial ICU hours
	Additional ICU hours
	Total hours ICU
Organism Attribute ∩ Quantitative Concept	Weight, height
Population Group ∩	Smoker
Finding ∩ Quantitative Concept	Smoker current
Idea or Concept	Urgent reason
	Emergent reason
Sign or Symptom	Insufficient Aortic
	Insufficient Mitral
Therapeutic or preventive procedure	Aortic procedure
	Mitral procedure

have been enriched through semantic enrichment by showing their categories and semantic types of STS and ACC. The STS terminologies contains 248 concepts and 32 categories. In the enriched terminology there are only 244 concepts, as we did not find semantic types for 4 concepts. The ACC terminology contains 142 concepts and 44 categories. The enriched ACC contains all 142 terms. The ACC and STS have neither attributes nor relationships. Thus, our integration is based on the names of concepts only.

In Table 8, we characterize the assignment of concepts to semantic types before and after running the ALSER algorithm. The left part of the table shows the expected increase in the number of semantic types due to ALSER, from 35 to 54 (for ACC) and from 38 to 68 for STS. Of these 54, 20 are intersection types, 26 simple semantic types and eight empty semantic types. The right part shows the distribution of concepts over semantic types before and after running ALSER. As expected, the number of concepts assigned to each semantic type is reduced for minimum, maximum, and average.

Table 9 (left part) shows results for running SEMINT with similarity threshold, α (90%). In an integrated ontology, every matched concept will occur only once. This is shown in the last two rows of the table. The right part of Table 9 shows more properties of the integrated TKB. The number of simple semantic types

Table 6
Semantic assignment in partial STS terminologies before/after semantic enrichment

Concept name	Category (before SE)	Semantic type (after SE)
Initial ICU hours	Post operative	Temporal concept
Additional ICU hours	Hospitalization	Temporal concept
Readmission to ICU	Hospitalization	Health care activity
Date of Birth	Demographics	Finding
Weight	History and risk factors	Organism Attribute ∩ Quantitative Concept
Smoker	History and risk factors	Population Group ∩ Finding ∩ Quantitative Concept
Patient ID	Administrative	Idea or concept
Urgent reason	Operative	Idea or concept
Patient last name	Demographics	Intellectual product
Insufficient aortic	Diagnostic cath procedure-findings	Sign or symptom
Aortic procedure	Operative	Therapeutic or preventive procedure

Table 7
STS and ACC concepts and categories (before semantic enrichment) and their semantic types (after semantic enrichment)

STS concept	STS category	ACC concept	ACC category	Semantic type
Cardiogenic shock Arrhythmia	Pre-operative cardiac status	Cardiogenic shock Arrhythmia	Diagnostic Cath Procedure—Indications	Disease or Syndrome ∩ Finding
MI	Pre-operative cardiac status	Previous MI	History and risk factors	Disease or Syndrome
Meds-Asprin	Pre-operative medications	Asprin	Cath Lab Visit—Medications	Organic Chemical ∩ Pharmacologic Substance
VD-Stenosis-Aortic VD-Stenosis-Mitral	Pre & Cath operative hemodynamics	Valve disease—Aortic Valve disease—Mitral	Diagnostic Cath Procedure—Findings	Disease or Syndrome ∩ Finding
RF-Renal Fail	Pre-operative risk factors	Renal failure	Adverse outcomes	Disease or Syndrome ∩ Finding

is 27. Looking back at the numbers of simple semantic types in Table 8, they are both 26. This indicates a high degree of overlap between the semantic types of ACC and of STS. There are 25 common semantic types, with only one semantic type that is unique to ACC (and to STS, respectively).

5.3. Recall and precision

We have performed a series of experiments to test the effectiveness of ALSER with SEMINT. We first computed a desired matching result between ACC and STS by hand. Then we used *recall* and *precision* to compare the results of ALSER with SEMINT with our manual result. We also compared with a brute force approach that matches every concept from ACC with every concept from STS. For evaluation purposes, we use the following common terms:

Table 8
Statistics of the effect of ALSER on ACC and STS

Feature	ACC	STS
Concepts#	142	244
Simple concepts	93	173
Compound concepts	49	71
Original semantic types	35	38
Refined semantic types	54	68
Intersection types	20	32
Simple semantic types	26	26
Empty semantic types	8	10

	ACC		STS	
Assigned to a semantic type	Bef.	Aft.	Bef.	Aft.
Min# of concepts	1	0	1	0
Max# of concepts	53	15	58	33
Avg.# of concepts	3	2	5	3
Total# of assignments	192	142	325	244

Table 9
(a) Matching results with threshold > 90; (b) Integrated terminology

Feature	Threshold $\alpha > 90$
(a)	
Matched concepts	37
ACC unmatched concepts #	105
STS unmatched concepts #	207
Simple concepts #	349
Total # of ass. to semantic types	349
(b)	
Features	Number
Simple semantic types #	27
Intersection types #	41
Empty semantic types #	15
Total semantic types	83
Minimum # of concepts assigned to a semantic type	0
Maximum # of concepts assigned to a semantic type	33
Average # of concepts assigned to a semantic type	3

True Positives (TP): The matching algorithm reports a match of two terms, and those two terms are considered a match according to a human expert.

True Negatives (TN): The matching algorithm does not report a match of two terms, and it should not report it, according to a human expert.

Table 10
Comparison of ALSER/SEMINT with brute force matching

Method	Threshold (%)	Precision	Recall	F measure	Time (in ms)
ALSER with SEMINT	> 70	0.925	0.8604	0.8915	551
ALSER with SEMINT	> 80	0.9736	0.8604	0.9135	551
ALSER with SEMINT	> 90	1	0.8604	0.925	551
Brute force	> 70	0.7916	0.8837	0.8351	16 543
Brute force	> 80	0.9268	0.8837	0.9047	16 543
Brute force	> 90	1	0.8604	0.925	16 543

False Positives (FP): The matching algorithm reports a match of two terms, but a human expert would not consider those two terms a match.

False Negatives (FN): The matching algorithm does not report two terms as matched, but they should be matched, according to a human expert.

We can now use Eq. (11) to compute the precision (P) and Eq. (12) to compute the recall (R) of our experiment. Low recall means that our algorithm is missing many pairs that it should report, according to our human expert. Low precision means that our algorithm is reporting many pairs that it should not report. An additional measure that is commonly used in Information Retrieval is Van Rijsbergen's F measure:

$$P = \frac{TP}{TP + FP}, \tag{11}$$

$$R = \frac{TP}{TP + FN}, \tag{12}$$

$$F = \frac{2 * R * P}{R + P}. \tag{13}$$

Table 10 shows the threshold α, P, R, F and runtime for ALSER with SEMINT and for brute force, given three different threshold values. Using ALSER/SEMINT, increasing the threshold to 90% leads to better precision, without deterioration of recall. Brute force is also better for $\alpha = 90\%$, as shown by the F value, although recall is slightly better for lower α. Thus, we concentrate on comparing the 90% entries in the table, which have the same P and R, but the run time is about 30 times faster for ALSER with SEMINT.

In future work, we are expecting better precision and F results for the ALSER with SEMINT method, hopefully without degradation of recall, because the current implementation does not make use of attributes and relationships, as these are not available for the ACC and the STS. Thus, matching relies on a very weak method. We will therefore investigate the use of better testbed ontologies. With larger ontologies, the runtime differences become more important.

6. Related work

Many lines of research have addressed ontology matching in the context of ontology construction and integration [23–26]. The major goal of these approaches is to develop effective methodologies for automated mappings [27]. A linchpin of the ALSER methodology is that it defines a notion of similarity. There is an extensive literature on similarity measures in ontologies, which can be effectively used in ontology integration.

As a major line of similarity measurement research, a number of linguistics researchers have attempted to make use of conceptual distance in information retrieval. These approaches mainly focus on a measure of semantic relatedness using WordNet and medical terminologies such as MeSH, UMLS, etc. Although these terminologies are quite thorough in their coverage of concepts, the number of semantic relation types connecting these concepts is considerably limited for advanced feature-based similarity models. Also relations in terminologies tend to be lexical relations between words, which do not always reflect ontological relations between classes of entities of the world. Tversky's feature-based similarity measure [28] is one of the most powerful similarity models, but it requires a rich knowledge base. A number of directions based on Tversky's work have been explored. Most approaches [29–31] have limited their attention to IS-A links and *broader term* and *narrower term* links.

A similarity measure using edge-counting [29] relies on decomposing concepts by explicit features and inherited features and ranks concepts by counting feature differences. Lee et al. [31] extended this method by including the links between synsets or paths made up of meronym type links.

When using the edge distance between two concepts, the relative depth is important, because research [32] has shown that the distance between concepts shrinks as one descends down the ontology. Two siblings near the top of the ontology are conceptually far apart, compared to two siblings deep down in the ontology. Relative depth was taken into consideration by [33]. Another important similarity issue is related to ontology density. Concept distances in denser regions should be smaller than distances in less dense regions. Hirst and St-Onge [34] also consider the direction and stability of links between concepts. In the HCG weighted conceptual distance methodology [35] the strength of connotation between parent and child nodes was considered together with ontology density and depth. Some research has combined thesauri with a corpus statistics for semantic similarity measurement [36–38].

Conceptual similarity is considered in terms of class similarity by Resnik [39]. In this approach, the extent of shared information between concepts was considered to determine their similarity. Lin [37] proposed an information-theoretic notion of similarity based on the joint distribution of properties. They accounted not only for commonalities but also for differences between the items being compared. In [40], a subclass must be of the same type as its parent, but must have some difference that distinguishes it from its parent. The subclasses of a concept are incompatible with each other. Jiang and Conrath's [38] similarity measurement is based on the conditional probability of encountering a child synonym set given a parent synonym set.

More recently, a number of new similarity approaches has been introduced. One approach is the use of matching rules [25,26,41]. Another method compares the set of all possible correspondences [42–45]. The names of the concepts or nesting relationships between concepts and the inter-relationships between concepts (slots of frames in [25]) are also criteria for comparison. The types of the concepts, or the labeled graph structure of the models [42,43] may be used to estimate the likelihood of data instance correspondences [41,44,46,47]. Rodriguez and Egenhofer [48] 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. Like in our approach, these three aspects are combined, using a weighted sum function. However, our approach is substantially different because of our use of semantic refinement prior to any matching. Furthermore, we do not use synonyms and distances; our term matching approach is based on bigrams.

Some similarity approaches [24,25] allow for efficient user interaction or expressive rule languages [23] for specifying mappings. Several recent publications have attempted to further automate the ontology matching process. A general heuristic was used in [49] to show that paths between matching elements tend to contain other matching elements. In [44], similarity between two nodes was computed based on their signature vectors, which were computed from data instances. The above approaches argue for a single best universal similarity measure, whereas GLUE [45] allows for application-dependent similarity measures.

A methodology for learning to construct ontologies from other ontologies is described in [27]. The similarity measure in [50] is based on statistics, and can be thought of as being defined over the joint probability distribution of the concepts involved. The role of machine learning in the semantic Web effort was described in [51]. Doan et al. [45] is another example of ontology integration using machine learning.

We now return to the problem of combining multiple ontologies into a single coherent ontology [25,41]. There are a few different methodologies but there is no consensus on the methodology to follow to integrate ontologies either in the same or in different domains. It is not even clear whether there could be a domain-independent methodology. When the semantics of an ontology integration system (whose sources are described by different ontologies) are defined, a global mediated ontology [42] might be required, whereas in some frameworks [52] a mediated ontology will exist only if it makes sense for the task at hand.

Pinto and Martins [53] addressed important issues of ontology engineering and ontology composition. Gangemi et al. [54] developed a domain ontology by the integration of existing repositories of knowledge, analyzing, and selecting the relevant sets of terms from various terminological sources. While [54] starts with all ontologies at the beginning of the merge process, Skuce [55] starts with a selected initial group of ontologies that is incrementally enlarged. The work of [42] describes a specialization of a framework for ontology integration.

7. Conclusion and future work

Matching concepts from two different medical ontologies is a difficult task, which will become more and more important with the expected advancement of medical informatics. Our approach to concept matching is based on an ontology representation with a TKB which makes it possible to identify concepts with similar semantics before attempting the difficult matching task. The semantic enrichment method builds TKBs and the algorithmic semantic refinement (ALSER) algorithm generates semantically uniform TKBs, for which it is possible to define a formal sense of semantic similarity between concepts.

Our approach is characterized by a philosophical assumption about future ontologies which will need to be integrated. Any ontology that will be important enough to be used at all will need to be "reasonably" complete in at least one subarea of a domain. Thus, if two ontologies cover a domain, one may expect that all the "important" concepts of the domain occur in both ontologies. Thus, a matching algorithm should attempt to maximize the number of matching concepts, even if there are structural differences between them. The semantic integration (SEMINT) algorithm performs this kind of matching on semantically uniform TKBs.

We presented an implementation of ALSER with SEMINT, applying it to the ACC and STS terminologies. Results of the implementation application were expressed in terms of precision, recall and F measure. ALSER with SEMINT greatly improves the run time of matching. It has a positive effect on precision for low thresholds and a negative effect on recall. Overall, as the F value shows, the improvement of precision outweighs the effect on recall.

This paper presents the beginning of a journey, not the end. At this point, we discuss some limitations of our approach. For research on integration, two TKBs in the same domain are necessary, which have to be built by independent teams, to avoid any biases that would make integration unfairly easier. With increasing growth of ontologies, we hope that more and more medical ontologies are available, which can be used in our research. For the case, when a global TKB ontology (i.e., UMLS) is available, we have presented a method for semantic enrichment to create such two-level structures for two medical terminologies of STS and ACC. In general, there could exist some newly generated or empty semantic types as the result of ALSER. Our current testbed ontologies have no attributes and no relationships. We expect better results by adding these features. Also, our method relies on a threshold value and several weights. Tuning these values and analyzing their effect on precision and recall of SEMINT is a topic of future research. Lastly, too little is known about TKBs themselves. We will investigate structural parameters of TKBs, foremost the ratio of the size of the semantic Network relative to the size of the thesaurus, to see what influence they have on running the ALSER and SEMINT algorithms.

References

[1] A. Rector, Clinical terminology: why is it so hard?, Methods Inf. Med. 38 (1999) 239–252.

[2] A. Rector, S. Bechhofer, C. Goble, I. Horrocks, W. Nowlan, W. Solomon, The GRAIL concept modelling language for medical terminology, Artif. Intell. Med. 9 (1997) 139–171.

[3] Unified Medical Language System Website. ⟨http://umlsks4.nlm.nih.gov⟩.

[4] T.R. Gruber, Toward principles for the design of ontologies used for knowledge sharing, in: N. Guarino, R. Poli (Eds.), Proceedings of International Workshop on Formal Ontology, Padova, Italy, 1993.

[5] J. Geller, H. Gu, Y. Perl, M. Halper, Semantic refinement and error correction in large terminological knowledge bases, Data Knowledge Eng. 45 (1) (2003) 1–32.

[6] H. Gu, Y. Perl, J. Geller, M. Halper, L. Liu, J.J. Cimino, Representing the UMLS as an OODB: modeling issues and advantages, J. Am. Med. Inf. Assoc. 7 (1) (2000) 66–80.

[7] American college of cardiology website. ⟨http://www.acc.org/⟩.

[8] Society of thoracic surgeon website. ⟨http://www.sts.org/⟩.

[9] B.L. Humphreys, D.A.B. Lindberg, Building the unified medical language system, in: Proceedings of the 13th Annual Symposium on Computer Applications in Medical Care, Washington, DC, November 1989, pp. 475–480

[10] B.L. Humphreys, D.A.B. Lindberg, H.M. Schoolman, G.O. Barnett, The unified medical language system: an informatics research collaboration, J. Am. Med. Int. Assoc. 5 (1) (1998) 1–11.

[11] D.A.B. Lindberg, B.L. Humphreys, A.T. McCray, The Unified Medical Language System, Methods Inf. Med. 32 (1993) 281–291.

[12] P.L. Schuyler, W.T. Hole, M.S. Tuttle, D.D. Sherertz, The UMLS metathesaurus: representing different views of biomedical concepts, Bull. Med. Libr. Assoc. 81 (2) (1993) 217–222.

[13] M.S. Tuttle, D.D. Sherertz, N.E. Olson, M.S. Erlbaum, W.D. Sperzel, L.F. Fuller, S.J. Nelson, Using meta-1 the first version of the UMLS metathesaurus, in: Proceedings of the 14th Annual SCAMC, 1990, pp. 131–135.

[14] A.T. McCray, UMLS semantic network, in: Proceedings of the 13th Annual SCAMC, 1989, pp. 503–507.

[15] A.T. McCray, Representing biomedical knowledge in the UMLS semantic network, in: N.C. Broering (Ed.), High-Performance Medical Libraries: Advances in Information Management for the Virtual Era, Meckler, Westport, CT, 1993, pp. 45–55.

[16] A.T. McCray, W.T. Hole, The scope and structure of the first version of the UMLS semantic network, in: Proceedings of the 14th Annual SCAMC, 1990, pp. 126–130.

[17] Y. Peng, M. Halper, Y. Perl, J. Geller, Auditing the umls for redundant classifications, in: Proceedings of AMIA 2002, San Antonio, TX, 2002, pp. 612–616.

[18] D. Waltz, Understanding line drawings with shadows, in: P. Winston (Ed.), The Psychology of Computer Vision, McGraw-Hill, New York, 1975, pp. 19–91.

[19] C.L. Hwang, K. Yoon, Multiple Attribute Decision Making, Springer, Berlin, 1981.

[20] K. Supekar, ALSER/SEMINT Implementation Report, 2002.

[21] D.F. D'Souza, A.C. Wills, Objects, Components, and Frameworks with UML: the catalysis approach, Addison-Wesley, Reading, MA, 1999.

[22] SAX website. ⟨http://www.saxproject.org/⟩.

[23] H. Chalupsky, Ontomorph: a translation system for symbolic knowledge, in: Principles of Knowledge Representation and Reasoning, Morgan Kaufmann, Los Altos, CA, 2000.

[24] D. McGuinness, R. Fikes, J. Rice, S. Wilder, The CHIMAERA ontology environment, in: Proceedings of the 17th National Conference on Artificial Intelligence, 2000.

[25] N. Noy, M. Musen. Prompt: algorithm and tool for automated ontology merging and alignment, in: Proceedings of the National Conference on Artificial Intelligence, 2000.

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

[27] A. Maedche, A machine learning perspective for the semantic web, in: Proceedings of Semantic Web Working Symposium (SWWS), 2001.

[28] A. Tversky, Features of similarity, Psychol. Rev. 84 (1977) 327–352.

[29] R. Rada, H. Mili, E. Bicknell, M. Blettner, Development and application of a metric on semantic nets, IEEE Trans. Syst. Man Cybernet. 19 (1) (1989) 17–30.

[30] Y.W. Kim, H. Kim, A model of knowledge based information retrieval with hierarchical concept graph, J. Doc. 46 (2) (1990) 113–137.

[31] J.H. Lee, M.H. Kim, Y.J. Lee, Information retrieval based on conceptual distance in IS-A hierarchies, J. Doc. 49 (2) (1993) 188–207.

[32] P.W. Lord, R.D. Stevens, A. Brass, C.A. Goble, Investigating semantic similarity measures across the gene ontology: the relationship between sequence and annotation, Bioinformatics 19 (2003) 1275–1283.

[33] M. Sussna, Word sense disambiguation for free-text indexing using a massive semantic network, in: Proceedings of the Second International Conference on Information and Knowledge Management (CIKM-93), 1993, pp. 67–74.

[34] G. Hirst, D. St-Onge, Lexical chains as representations of context for the detection and correction of malapropisms, in: M. Hoey (Ed.), Patterns of Lexis in Text, Oxford University Press, Oxford, 1998, pp. 305–332.

[35] E. Agirre, G. Rigau, Word sense disambiguation using conceptual density, in: Proceedings of the 16th International Conference on Computational Linguistics, 1996, pp. 16–22.

[36] P. Resnik, Using information content to evaluate semantic similarity, in: Proceedings of the 14th International Joint Conference on Artificial Intelligence, 1995, pp. 448–453.

[37] D. Lin, An information-theoretic definition of similarity, in: Proceedings of the 15th International Conference on Machine Learning, 1998.

[38] J. Jiang, D.W. Conrath, Semantic similarity based on corpus statistics and lexical taxonomy, in: Proceedings of International Conference on Research in Computational Linguistics, 1997.

[39] P. Resnik, Selection and information: a class based approach to lexical relationships, Ph.D Thesis, University of Pennsylvania, 1993.

[40] J. Bouaud, B. Bachimont, J. Charlet, P. Zweigenbaum, Methodological principles for structuring an ontology, in: IJCAI-95 Workshop on Basic Ontological Issues in Knowledge Sharing, 1995.

[41] G. Stumme, A. Maedche, FC A-MERGE: bottom-up merging of ontologies, in: Proceedings of the 17th International Joint Conference on Artificial Intelligence (IJCAI), 2001.

[42] D. Calvanese, D.G. Giuseppe, M. Lenzerini, Ontology of integration and integration of ontologies, in: Proceedings of the 2001 Description Logic Workshop (DL2001), 2001.

[43] S. Melnik, H. Molina-Garcia, E. Rahm, Similarity flooding: a versatile graph matching algorithm, in: Proceedings of the International Conference on Data Engineering (ICDE), 2002.

[44] M. Lacher, G. Groh, Facilitating the exchange of explicit knowledge through ontology mappings, in: Proceedings of the 14th International FLAIRS Conference, 2001.

[45] A. Doan, J. Madhavan, P. Domingos, A. Halevy, Learning to map between ontologies on the semantic web, in: Proceedings of WWW-2002, 2002.

[46] A. Doan, P. Domingos, A. Halevy, Reconciling schemas of disparate data sources: a machine-learning approach, in: Proceedings of SIGMOD, 2001, pp. 509–520.
[47] J. Berlin, A. Motro, Database schema matching using machine learning with feature selection, in: Proceedings of the 14th International Conference on Advanced Information Systems Engineering (CAiSE02), 2002.
[48] M.A. Rodriguez, M.J. Egenhofer, Determining semantic similarity among entity classes from different ontologies, IEEE Trans. Knowl. Data Eng. 15 (2) (2003) 442–456.
[49] N. Noy, M. Musen, Anchor-PROMPT: using non-local context for semantic matching, in: Proceedings of the Workshop on Ontologies and Information Sharing at the International Joint Conference on Artificial Intelligence (IJCAI), 2001.
[50] R. Ichise, H. Takeda, S. Honiden, Rule induction for concept hierarchy alignment, in: Proceedings of the Workshop on Ontology Learning at the 17th International Joint Conference on Artificial Intelligence (IJCAI), 2001.
[51] A. Maedche, S. Saab, Ontology learning for the semantic web, IEEE Intell. Syst. 16 (2) (2001).
[52] W.R. Swartout, R. Patil, K. Knight, T. Russ, Towards distributed use of large-scale ontologies, in: AAAI Spring Symposium on Ontological Engineering, 1997.
[53] H. Sofia Pinto, J.P. Martins, A methodology for ontology integration, in: Proceedings of the First International Conference on Knowledge Capture (K-CAP2001), 2001.
[54] A. Gangemi, G. Steve, F. Giacomelli, ONIONS: an ontological methodology for taxonomic knowledge integration, in: ECAI Workshop on Ontological Engineering, 1996.
[55] D. Skuce, How we might reach agreement on shared ontologies: a fundamental approach, in: AAAI Spring Symposium Series, Workshop on Ontological Engineering, 1997.

Yugyung Lee is an Assistant Professor at the University of Missouri at Kansas City. She received a B.S. in Computer Science from the University of Washington in 1990 and a Ph.D. in Computer and Information Sciences from the New Jersey Institute of Technology in 1997. Before joining UMKC, she was working at MCC. She is Director of the UMKC Distributed Intelligent Computing (UDIC) Lab at the School of Computing and Engineering at UMKC. Her research interests include Medical Informatics, Knowledge Representation and Reasoning, Ontologies, Distributed Computing and Software Architectures, Middleware, Pervasive Computing and Context-aware System and Applications, Semantic Web and Web Services.

Kaustubh Supekar is a first year Ph.D. student in the Department of Biomedical Informatics at Stanford University. He received a BEng degree in Computer Engineering from the University of Mumbai, India and a Masters degree in Computer Scinece from the University of Missouri, Kansas City. His research interests include Computational Knowledge management, Medical Imaging, Semantic Web services, Peer-to-Peer and Biomedical Ontologies.

James Geller received an Electrical Engineering Diploma from the Technical University, Vienna, Austria, in 1979, and the MS Degree (1984) and Ph.D. degree (1988) in Computer Science from the State University of New York at Buffalo. He joined NJIT in 1988, was granted tenure in 1993 and promoted to full professor in 2000. He is Director of the Semantic Web and Ontologies Lab at the CS Department and Associate Chair for Undergraduate Studies. Previously, James Geller was Codirector and Graduate Advisor of the Biomedical Informatics M.S. and Ph.D. Programs jointly administered with UMDNJ. He has published in numerous journals and conferences on topics in Artificial Intelligence, Knowledge Representation, Parallel Reasoning, Databases, Semantic Modeling in Object-Oriented Databases, Medical Informatics, Medical Vocabularies, and Auditing of Medical Terminologies. His work on Medical Vocabularies (together with Y. Perl) was funded by the National Institute of Standards and Technology with over one million dollars. His work on Web Mininig for Marketing was supported by a five year grant of the NJ Commission of Science and Technology. Recent topics of interest include Information Integration and E-Commerce.