

Federated Ontology Search for the Medical Domain

Vasco Calais Pedro¹ and Lucian Vlad Lita² and Stefan Niculescu² and Bharat Rao² and Jaime Carbonell¹

¹ Language Technologies Institute,
School of Computer Science,
Carnegie Mellon University

² Siemens Medical Solutions ADDRESS PLACEHOLDER

Abstract. In this paper we describe a novel methodology for retrieving and combining information from multiple ontologies for the medical domain. In the last decades the number and diversity of available ontologies for the medical domain has grown considerably. The variety and number of such resources available makes the cost to integrate them into an application incremental, often prohibitive for exploratory prototyping, and discouraging for larger-scale integration. Cross-ontology localized merging is proposed as a way to allow for a flexible and scalable solution. This approach also indicates a low maintenance cost and high reusability for different application types within the medical domain

1 Introduction

In the last decades the number of available medical ontologies has grown considerably. Several proprietary and public efforts such as MESH [11] and SNOMED [18] have become available. Even though it is considered a concrete domain, the quantity of concepts and the diversity of semantic types typically included in the medical domain increases its complexity to levels close to an open domain ontology. This suggests that the problems found in dealing with an open domain ontology are also present when dealing with the medical domain. Some evidence of this is given by the present number of ontologies, taxonomies, and dictionaries found within the medical domain. A comparison of available ontologies reveals both redundant and complementary coverage, but the variety of frameworks and languages used for ontology development makes it a challenge to merge query results from different ontologies. The number of available languages for ontological knowledge engineering such as RDF, OWL, DAML+OIL, and CYCL, combined with the existence of independent interfaces aggravates the issue. The lack of a formal way to access and combine the knowledge from different ontologies is an obstacle to more effective re-use and combination of these medical resources.

One approach to the multi-ontology issue is to absorb all the knowledge across sources into a common ontology typically casting it into a single format. The most relevant instance of this approach in the medical domain is UMLS [10] (Section 3.2). However this approach has several drawbacks, as defined in

[17, 9], such as (i) non-scalability, (ii) losing language and reasoning specificity of distinct ontologies, (iii) losing privacy and autonomy of ontological knowledge (iv) language level mismatches such as syntax mismatches, differences in logical representation and different semantic primitives and (v) Ontology-level mismatches, such as difference in scope, coverage and granularity, making this challenge thus far too daunting in practice. Another approach is to punt on the merging problem and simply query more than one ontology via different interfaces, interpreting the results of each ontology individually, essentially moving the entire challenge from the ontology provider to the application builder. A third approach [13] is to build an ontological middleware level for only small fragments of ontologies in an on-demand basis:

- Query multiple ontologies and merge the query results from multiple knowledge base systems, much like *federated search* in information retrieval [4].
- Follow ontological chains and inferences across ontologies, using partial query results from one ontology to query another. This is a more complex version of cross-database joins, where the data schemas are sufficiently compatible.

As described in [8], currently, the main approaches to a solution for these problems focus on ontology integration, by creating a mapping between the concepts and relations of different ontologies. In some cases, such as the Semantic Web project [2] the focus lies primarily on merging ontologies by establishing a full mapping among them.

Some efforts have tried to produce a merged ontology automatically using a bottom-up approach such as FCA-Merge [19]; most involve some degree of semi-supervised mapping. Other approaches, such as the one taken by CYC, try to absorb other ontologies into a single main ontology while maintaining coherence [15]. One disadvantage of these approaches is the prohibitive cost of producing a mapping or of absorbing an ontology, given their increasing scale and rate of availability. Another disadvantage is that it is not always possible to establish a one-to-one mapping between the concepts and relations in one ontology and the concepts and relations in another. Furthermore, there is the problem of keeping the mappings updated as the original ontologies evolve. A large number of available ontologies are considered works in progress and are updated frequently, which implies a constant updating of any mappings associated with those resources. Finally, the problem of the profitability of ontological creation is particularly important in the medical domain. Medical knowledge is generally produced by experts in the medical field, which are often very expensive resources. In order to maintain updated information in a particular medical field, it is necessary to create viable business models for ontological knowledge creation.

Most applications that use medical ontological information would benefit from an approach that models the information need, queries the relevant ontologies, and retrieves the best result while providing a single unified interface to the client application. Towards this end we propose *Federated Ontology Search* (FOS). Similar to federated search [4, 6], in FOS we model ontologies as individual sources, construct queries that describe the information need, execute them

on each ontology independently, and finally merge the results into a single ranked list. Using Federated Ontology Search we can parallelize query execution while respecting the structure of the individual ontologies, taking advantage of both redundant and complementary knowledge in the available ontologies to improve the overall performance of the system.

XXXX Briefly describe what comes next

2 Approach

2.1 Ontological Search

The success of the proposed approach hinges on the definition of a search method that is independent of any particular ontology. For this purpose we introduce the concept of operator and a concept of query based on operators. The main purpose of an operator is to decouple the search process from the information need. Instead of describing a complete semantic framework, the goal is to describe the information request in terms of a decomposable query that can be transformed into a set of operators. This would provide an elegant abstraction from the formal representations implemented by our ontological sources, allowing each operator to be an independent request.

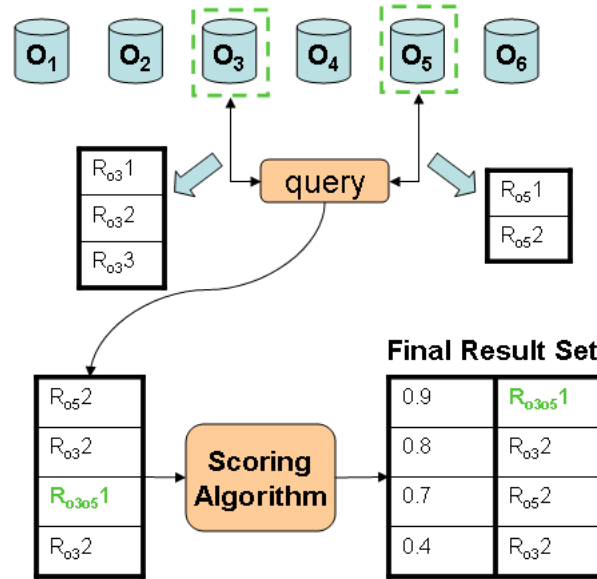


Fig. 1. Query operation in the Federated Approach

As an example, Figure 1 shows the execution of a query. It is important to note that by defining a set of operators we are in fact delegating responsibility for their execution to the ontologies themselves, therefore making no restrictions on whatever processes are executed in order to obtain the necessary information. This means that operators can be implemented using extended features of ontologies (e.g. inference, grounding, restrictions and theorem-provers). The only constraint is that the output of each query execution is a Rooted Directed Acyclic Graph (RDAG).

Operators We proceed to describe the set of operators that were implemented in our current prototype, as given by this simplified description for purposes of clarity;

- $\text{rel}(a,b,\text{rels})$ The relation operator returns a graph containing a chain linking a and b using the relations contained in rels .
- $\text{parents}(a)$ Returns the graph containing the parents of a .
- $\text{children}(a)$ Returns the graph containing the children of a .
- $\text{syn}(a)$ Returns the synonyms of a .

All operators must have the property of compositionality in order to enable the constructions of complex operations based on simpler ones. The ideal operator set should contain the operations usually performed in ontologies. Providing the full set of operators is not the scope of this paper but rather future work.

There are three sub-problems described in the literature on Federated Search that also apply to this area: resource description, resource selection and results merging and scoring. In the ontological domain, resources are ontologies.

2.2 Ontology Description and Selection

The goal is to select a subset of the available ontologies based on some measure of appropriateness of an ontology given a query. This can be modeled as the probability $p(O, q)$, where O is an ontology and q is a query.

Distributed online efforts such as OpenMind result in ontologies that are constantly updated thus making the task of modeling them very difficult. We must also consider cases where the ontologies make use of inference engines and logic mechanisms.

Therefore, a more general approach is to evaluate the relative utility of different ontologies by comparing the results they produce for a given input query. Although theoretically we could create a gold set of queries to perform on each ontology and use the results to create an ontology model, due to the diverse nature of ontologies this is in practice a very hard task. At this point we use a naive approach where the query is executed in the subset of ontologies that contain the concepts in the query, but this can bring problems in both performance and accuracy. In order to mitigate this risk, we created a parameter that models the general accuracy for a given source. This parameter is a good candidate for the application of a learning algorithm such as some variation of the Expectation

Maximization algorithm [12]. We need to create gold standards to train and test such algorithms, which makes the process time consuming.

2.3 Merging and Scoring

One advantage of this approach is that rather than trying to merge two entire ontologies, we merge only the results. This significantly reduces the problem of merging ambiguous concepts given that queries are grounded in a query concept and therefore only relations that apply to the concept in the query are returned. Polysemic concepts will be explicitly modeled via separate results, with a variable confidence on the relations contained in each result.

The structured nature of the query results implies that the result merging problem in the ontology domain is very different from result merging in federated text search, where the results typically consist of unstructured data. In the case of Ontology Search the results are graphs, which allow us to treat merging as an instance of Inexact Graph Merging. While one of the goals of merging in Federated Search is to eliminate duplicates, the primary goal of merging in ontology search is to find complementary information. The goal of ranking in ontology search is to produce, at the top-ranked position, the most complete and accurate result. Generally speaking, we want to merge two results if they represent information about the same concepts, thus creating a more complete result.

As an example, let us consider three distinct ontologies, O_1 , O_2 and O_3 . Let us assume that we execute the query $sim(bank)$, with the purposes of finding concepts similar to $bank$. Imagine that ontology O_1 and O_2 both interpret $bank$ as $river_bank$ while ontology O_3 interprets $bank$ as $money_bank$. Given this scenario, we would like the result set to contain two results, one referring to the similar concepts of $river_bank$ and the other to the similar concepts of $money_bank$. The results from O_1 and O_2 should be merged since that is likely to yield an increase in the quantity of information contained in the result, as well as the confidence in the concepts common to the two results.

The problem can be formulated as follows. Given two RDAGs g_1 and g_2 we want to merge the two graphs if they are similar. Typically this is done by considering one of two options, either we measure the similarity between the graphs or measure the difference. But in our case we not only want to measure the similarity but also find the maximum common subgraph. We should look to the science of Inexact Graph Matching for guidance here.

Inexact Graph Matching occurs when we do not expect to find an isomorphism between the two graphs to be merged. This is one of the most complex problems in computer vision [3] and is also an important part of chemical similarity searching [14]. More specifically, inexact graph matching is proven to be an NP-Complete problem [1].

In the next section we will discuss graph similarity.

Graph Similarity Graph similarity Distance [16] is typically calculated in one of the following ways: Cost Based Distance, Feature Based Distance or Maximum Common Subgraph.

Cost Based Distance is based on edit operations on the graph, typically add nodes or edges, remove nodes or edges and re-label nodes or edges, where each operation is associated with a cost. Given two graphs $g1$ and $g2$, the edit distance between $g1$ and $g2$ is the minimum number of edit operations necessary to transform $g1$ into $g2$.

Feature based distances use a set of invariants established from the graph structural description, using these features in a vector representation to which we then apply distance or similarity measures.

The goal of the Maximum Common Subgraph approach is to find the largest Subgraph common to both $g1$ and $g2$. To address this requirement, current approaches use the concept of maximum clique detection. The problem is then changed into finding the Maximum Common Edge Subgraph. In our case we use a variation of the overlapping coefficient for graphs, a measure whereby if graph g contains g' or the converse then the similarity coefficient is a full match.

Localized Confidence Boosting Algorithm Given $g1$ and $g2$ as results of a query, the algorithm is as follows. After applying the screening procedure we are left with graphs where $sim(g1, g2) > T$. Given that we now basically have $g1 \cap g2$ we will apply localized boosting and then add the nodes and edges that were previously discarded.

The basic intuition behind the confidence boosting is that the confidence of the edges is boosted whenever two edges are merged. The boosting is determined through the use of the *SoftOr*, given by the formula:

$$1 - \prod_i (1 - C_i)$$

In order to apply confidence boosting we apply the concept of tuples, where $t_x = (c_x, c_y, r)$ is a tuple, c_x, c_y are concepts and r is a relation. First we split $g1$ and $g2$ into tuples $t_x = (c_x, c_y, r), c_x, c_y, r \in g$, such that c_x and c_r are adjacent and $r(c_x, c_y)$. We then compare the sets of tuples from $g1$ and $g2$ and if $sim(t_x, t_y) > T$ then we boost the confidence of t_x .

Tuple Similarity Tuple similarity measures are based on the linear combination of the edge similarity and the concept similarity.

When comparing concepts or relations, we use the Q-Gram distance [20] on the strings that represent them. The intuition behind the use of q-grams as a foundation for distance metric is that when two strings $s1$ and $s2$ are within a small edit distance of each other, they share a large number of q-grams in common. This metric is fairly robust to orthographic errors, morphological errors and compound words, which makes it suitable for our purposes. The similarity between two tuples is given by the minimum similarity of the concepts and relations contained in the tuples. Formally

$$sim(t_x, t_y) = \min \begin{cases} sim(c_{x1}, c_{y1}) \\ sim(c_{y1}, c_{y2}) \\ sim(r_x, r_y) \end{cases}$$

Scoring Results A result is scored in a compositional manner, by scoring the outcome of each operator used in a query individually before calculating the final score. A ranking will be computed from the scores of the results thus making the computation of the ranking score a key issue, much like in traditional Information Retrieval.

Each operator focuses on either recall or precision. Operators that focus on recall will typically return results with as much information as possible. The similarity operator, for example, returns all the synonyms associated with a concept, the more synonyms the better the result should be, all else being equal. Operators that focus on precision will usually return chains of associations. They focus on precision of the relations. As an example, the relation operator finds the relation between two concepts. All else being equal, a direct relation would be better than a long chain of relations.

When a result is merged from two other results, the confidence in the sources from where the results were extracted is combined using the soft or rule, as described before. We now present two scoring metrics, to be used by precision type operators and recall type operators respectively.

Precision scoring metric The goal of this scoring metric is to give preference to shorter graph lengths. Given that this metric is used for precision type operators, it is desirable to have results with short chains.

$$S(r) = C_s \frac{\prod_i(c_{ei})}{avg_length^k}$$

Where C_s is the confidence of the source, c_e is the confidence on the edge and avg_length is the average distance of the paths contained in the graph, from root to leaf node. The constant k is determined heuristically in order to obtain a smooth curve and we decided on $k = 4$.

Recall scoring metric The recall scoring metric gives preference to graphs with large node degrees. Given that this metric is used by recall type operators, we want the score to increase with the quantity of diverse information contained in the graph. Thus

$$S(r) = C_s \times \left(1 - \frac{1}{avg_degree^k}\right) \times \prod_i(c_{ei})$$

Where avg_degree is the average degree of the nodes contained in the result.

3 Characterization of Medical Resources

We can divide the available medical ontological resources according to several characterizing dimensions; Scope, Depth, Cost and Noise.

- Scope
 - Coverage is defined by the percentage of broad semantic topics covered by the ontology. Ontologies can vary according to coverage. The broader the coverage the more diverse is the ontology.
- Depth
 - Ontology Depth is defined by
 - * the average path length in the ontology. The higher the average path the more specialized is the ontology
 - * the number of concepts in each semantic type on average
 - * [we should somehow include relations into characterization]
- Cost
 - Cost is defined by $T_c \times C_c$, where T_c is the amount of time in human hours required per concept and C_c is the cost in dollars per concept. The range varies from fully automatic ontologies, which require almost no human effort, to specialized, fully manually generated, ontologies, which not only require a large amount of time per concept as well as a highly specialized (and therefore expensive) human to generate that knowledge.
- Noise
 - Noise is defined by the percentage of concepts in the ontology that carry no semantic meaning. Given a set of concepts, a human should be able to determine which concepts carry semantic meaning. This is especially important in automatically created ontologies that tend to introduce semantically irrelevant concepts due to the extraction process.

Any ontology can be categorized according to these three measures. Typically there is an inversely proportional relation between coverage and depth

In figure 2
In table 1

	Scope	Depth	Cost	Noise	Connectivity
Wordnet	medium-high	low	medium	low	medium
UMLS	low	high	high	low	low
Okinet	high	high	low	high	high

Table 1. Categorization Table

Although it is arguable the exact score of each ontology, this scale is meant to describe ontologies in a broad sense. Generally speaking it should be able to compare any two ontologies given these dimensions. We consider that there are possibly many characterization schemes for ontologies, although they are contradictory with the one proposed here. We now describe what we consider to be clear examples of different ontology types according to the characterization given above.

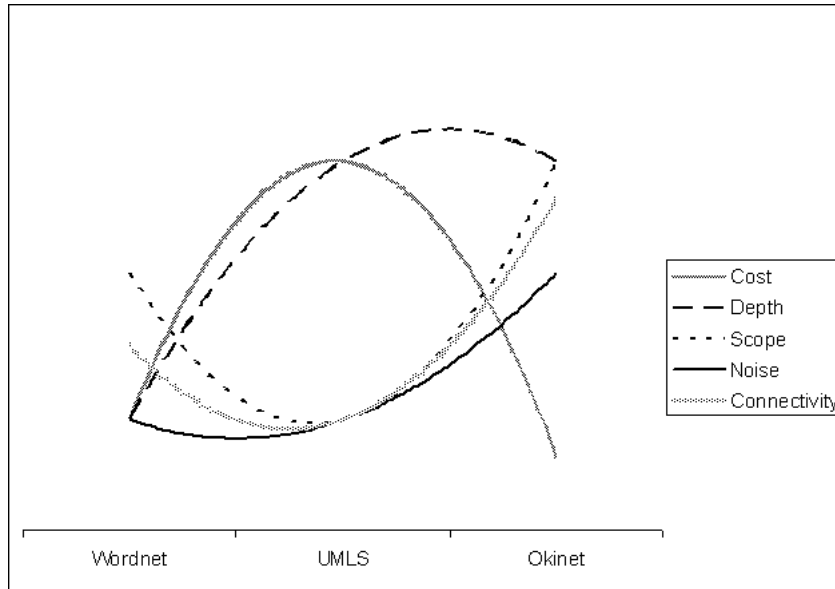


Fig. 2. Ontology Categorization

3.1 Wordnet

WordNet is a semantic lexicon for the English language. It is perhaps the most widely used ontology in the world. It groups English words into sets of synonyms called synsets, provides short definitions, and records the various semantic relations between these synonym sets. It currently holds 155327 unique strings. It is the canonical example of an ontology with high coverage, low depth and medium cost.

3.2 Unified Medical Language System

The National Library of Medicine’s Unified Medical Language System (UMLS) [7] is both an ontology and a set of ontologies. It aims to bring together a set of medical dictionaries (currently holding roughly 100) and contains information about medical concepts, semantic types, and the relations between concepts and semantic types. It is maintained mostly by hand and it is the result of a 15 year long project that involved many thousand of man hours (perhaps develop and make concrete affirmations backed by papers citation). It currently contains over 1 million concepts. UMLS represents the current largest effort of manual ontology mapping. Due to its specific nature and medical domain expertise, UMLS has low coverage (deals mostly with the medical domain), high depth (it is very specialized) and has a high cost (the nature of the knowledge it contains requires very specialized knowledge providers).

3.3 Okinet

Okinet is a Wikipedia [5] Based Ontology and is currently being developed at Siemens Research. It presents a unique set of challenges in the sense that is an ontology derived from a source that is updated very frequently. Any method other than a fully autonomous will carry significant difficulties in updating the ontology. At the same time, it highlights the difficulty of maintaining mappings between ontologies, given that not only the amounts of concepts included in wikipedia is enormous, but also that the update rhythm is extremely high. Therefore it aims to bring together a set of machine learning and natural language techniques to create an ontology fully automatically from wikipedia, without human supervision. In addition to information previously extracted in [citation to works that mine wikipedia] we use a combination of feature based and graph based approach to ontology building. It currently contains more than 4 million concepts and encodes a wide range of semantic relations from the more traditional such as synonyms, hypernyms, etc., as well as instance based semantic relations such as symptom_of, causes, medication, etc. It represents the opposite approach to UMLS, the automatic creation of an ontology from a semi structured document corpus. Okinet is high coverage (it is an open domain ontology), high depth(it is incredibly rich) and low cost (the knowledge exists already) ontology. One could argue that we should consider the cost of creating wikipedia when considering the cost of creating okinet, but given that the knowledge was not created specifically for okinet, it is not part of our calculations.

4 Current Work

The current focus is on integration of multiple ontologies in the medical domain for immediate use. We currently have integrated UMLS, Okinet and Wordnet as a proof of concept application. We have built a Ontology Search System that uses the methods described in this paper and produced a simple interface for ontology search. We call this integrated system ROSE, the Remind Ontology Search Engine.

4.1 UMLS issues

Even though UMLS an extremely rich and large ontology, it presents some challenges regarding certain types of relation types. UMLS provides two different knowledge sources, the Metathesaurus and the Semantic Network. The Metathesaurus establishes relations at the concept level and the Semantic Network describes the relations at the semantic type level. The problem is due to the fact that there is no direct correlation between the two in most cases. Even though some work has been done in mapping the two relation sets [enter citation here], there is currently not possible to retrieve information regarding semantic relations between concepts that belong to specific semantic types other than hierarchical information. E.g “what is the medications that treat disease X?” is a

question that is not possible to answer using UMLS. We aim to leverage the use of multiple ontologies to fill the gaps in UMLS, thus creating a very powerful ontological solution.

5 Web Based Interface

Although ROSE works in a typical server fashion, which can be queried directly using xml, we have also created a visualization tool for ontology browsing. Given the fact that the server is querying many ontologies, possible resources that are located in a different location, having a localized copy of everything is unfeasible. We therefore opted for the creation of a web interface to visualize the results of ontology queries.

The web interface allows for querying ROSE for synonyms from all the available ontologies. It enables the user to visualize the synonymy relations across ontologies in a graph display and browse subsequent relations with ease and simplicity. It uses AJAX and JSP with JSON as a communication protocol. Below is a snapshot of the result of querying rose for Congestive Heart Failure. We can see the results from Wordnet, UMLS and Okinet represented by different colors.

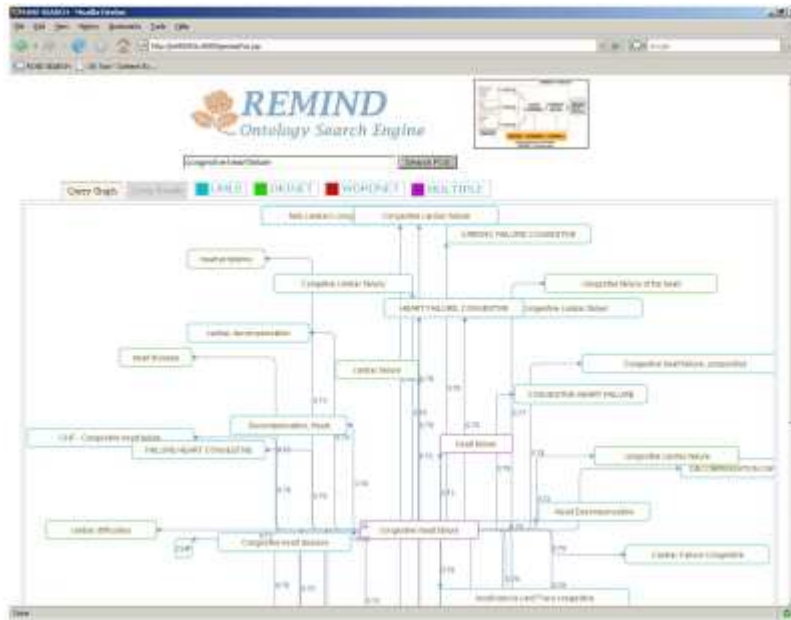


Fig. 3. Web Based Interface

6 Future Work

One of the main challenges is the association of confidences with ontology results. Given that most ontologies do not have confidences associated with the content, but rather encode knowledge as binary information type (either exists or not) determining which ontology or ontology set returns the best content is an open ended problem. We are addressing this with the creation of Okinet, which will contain confidence scores for each relation in the ontology, and we hope to use that as a basis for a default scoring procedure for ontological relations.

Reliance on string similarity for concept and edge matching presents a problem since it assumes that namespaces are shared amongst ontologies with a small degree of variation. Instead of resorting to manual mapping in the spirit of UMLS we are developing a method for leveraging the use of many ontologies to determine conceptual similarity.

7 Conclusions

In this paper we describe a novel methodology for retrieving and combining information from diverse ontologies in the medical domain. This methodology present several benefits over the current approaches since it maintains ontology independence, allows for easier maintenance, enables the existence of viable ontology creation models and addresses the issue of dynamic ontologies, in which the continuous update to the ontology makes it costly to adopt a merge approach. We establish a parallel to federated search in information retrieval and proceed to describe the algorithms for merging and scoring the results from multiple ontologies and we introduce a preliminary set of operators, currently implemented in a set of ontologies. Although there are many challenges to be solved, this is a promising approach that brings closer the goal of easier integration of information from different ontological sources.

References

1. A. M. Abdulkader. Parallel algorithms for labelled graph matching. 1998.
2. T. Bemers-Lee, J. Hendler, and O. Lassila. The semantic web. *Scientific American*, 284(5):34–43, 2001.
3. E. Bienenstock and C. von der Malsburg. A neural network for invariant pattern recognition. *Europhysics Letters*, 4:121, 1987.
4. J. Callan. Distributed information retrieval. advances in information retrieval, 2000.
5. C. Circuit. Wikipedia. 2005.
6. D. Fryer. Federated search engines. *Online(Weston, CT)*, 28(2):16–19, 2004.
7. B.L. Humphreys and D.A.B. Lindberg. The UMLS project: making the conceptual connection between users and the information they need. *Bulletin of the Medical Library Association*, 81(2):170, 1993.
8. Y. Kalfoglou and M. Schorlemmer. Ontology mapping: the state of the art. *The Knowledge Engineering Review*, 18(01):1–31, 2003.

9. M. Klein. Combining and relating ontologies: an analysis of problems and solutions. *Workshop on Ontologies and Information Sharing, IJCAI*, 1, 2001.
10. DA Lindberg, BL Humphreys, and AT McCray. The Unified Medical Language System. *Methods Inf Med*, 32(4):281–91, 1993.
11. C.E. Lipscomb. Medical Subject Headings (MeSH). *Bull Med Libr Assoc*, 88(3):0265–0266, 2000.
12. T. K. Moon. The expectation-maximization algorithm. *Signal Processing Magazine, IEEE*, 13(6):47–60, 1996.
13. Vasco C. Pedro, Nyberg Eric, and Jaime Carbonell. Federated Ontology Search. *Proceedings of the Semantic Information Integration on Knowledge Discovery*, 2006.
14. J. W. Raymond, E. J. Gardiner, and P. Willett. Rascal: Calculation of graph similarity using maximum common edge subgraphs. *The Computer Journal*, 45(6):631–644, 2002.
15. S. Reed and D. Lenat. Mapping ontologies into cyc. *AAAI 2002 Conference Workshop on Ontologies For The Semantic Web, Edmonton, Canada, July*, 2002.
16. A. Sanfeliu and K. S. Fu. A distance measure between attributed relational graphs for pattern recognition. *IEEE Transactions on Systems, Man, and Cybernetics*, 13:353–362, 1983.
17. L. Serafini and A. Taminin.
18. K.A. Spackman, K.E. Campbell, and R.A. Cote. SNOMED RT: a reference terminology for health care. *Proc AMIA Annu Fall Symp*, 640(4):503–512, 1997.
19. G. Stumme and A. Maedche. Fca-merge: Bottom-up merging of ontologies. *7th Intl. Conf. on Artificial Intelligence (IJCAI'01)*, pages 225–230, 2001.