Multi-Search: A Meta-search Engine Based on Multiple Ontologies

Mohammed Maree[1], Saadat M. Alhashmi[1], Mohammed Belkhatir[2],
Hidayat Hidayat[1], and Bashar Tahayna[1]

[1] Monash University, Sunway Campus, Malaysia
[2] University of Lyon & CNRS-Francesity, France
{mohammed.maree,saadat.m.alhashmi,hhid1,
bashar.tahayna}@infotech.monash.edu.my,
mohammed.belkhatir@iut.univ-lyon1.fr

Abstract. In this paper, we present *Multi-Search* meta-search engine. *Multi-Search* combines three approaches: meta search, ontology-based semantic translation techniques, and statistically-based semantic relatedness measures. *Multi-Search* attempts to employ knowledge represented by multiple ontologies for both query translation and returned results merging. In addition, it utilizes semantic relatedness measures to address the issue of missing background knowledge in the used ontologies. The developed system operates on top of several search engines and can be easily extended. Experimental results indicate that the techniques used to build the meta-search engine are both effective and efficient.

Keywords: meta-search, ontology, query translation, semantic relatedness.

1 Introduction

Current internet search engines have a number of deficiencies. First, these search engines still suffer from low precision/recall ratio [1]. The reason behind this is because these search engines use keyword-based indexing techniques to index WebPages. Although this approach assist users in finding information on the Web, many of the returned results are irrelevant to the user's intent. This is due to the "semantic-gap" between the meanings of the keywords that are used to index WebPages and the meanings of the terms used by the user in his query. Second, the web-coverage by a single search engine may be limited. A study conducted by [2] showed that the index intersection between the largest available search engines (Google, Yahoo!, MSN and ASK) is estimated to be 28.8%. Therefore, combing results returned from multiple search engines can be seen as an effective solution to this problem. In this paper, we introduce *Multi-Search,* a meta-search engine for retrieving, merging and ranking results returned by several individual search engines. The proposed system employs knowledge represented by multiple ontologies to derive the semantic aspects of both the user query and returned search results. In addition, statistically-based semantic relatedness measures are utilized to compensate for missing background knowledge in the exploited ontologies. In our approach, we believe that users must be considered as the center of the search process. Therefore, in *Multi-Search,* users can filter and rank

P.-J. Cheng et al. (Eds.): AIRS 2010, LNCS 6458, pp. 110–120, 2010.

the results by giving them weights according to their relevancy to the query intent. To overcome the low coverage problem, the proposed system operates on top of several search engines such as Google (www.google.com), Yahoo! (www.yahoo.com), Bing (www.bing.com), and it can be easily extended by plugging additional search engines. We summarize our contributions as flows:

- Unlike traditional keyword-based indexing approaches, *Multi-Search* employs knowledge represented by multiple ontologies to derive the semantic aspects of both user query and returned search results.
- *Multi-Search* combines semantic and statistical based techniques to compensate for missing background knowledge in the used ontologies.

The rest of the paper is organized as follows. Section 2 presents an overview of the related work. A general overview of the proposed meta-search engine is given in section 3. Section 4 explains in detail the proposed methods. Section 5 discusses experiments carried out to evaluate our meta-search engine. The final section presents the conclusions and outlines the future work.

2 Related Work

2.1 Ontology-Based Semantic Translation (OBST)

Ontologies play a crucial role in deriving the semantic aspects in both text and content-based information retrieval systems. Several OBST systems have been proposed but all of them either use a single ontology or multiple ontologies for a specific domain. Among the first systems that used ontology for this purpose is OntoSeek [3]. This system is designed for content-based information retrieval from online yellow pages and product catalogs. It uses the Sensus ontology which comprises a simple taxonomic structure of approximately 70,000 nodes to represent queries and resource descriptions. The system proposed by [4] uses subject hierarchies provided by online portals such as Yahoo.com and About.com as reference ontology for personalized web search. The authors of [5] propose to use multiple ontologies in specialized domains for information extraction purposes. Their experimental results show that by using multiple ontologies precision can be improved. In *Multi-Search*, we are not interested in a particular domain; therefore, we propose to use general-purpose ontologies that cover knowledge in multiple domains.

2.2 Meta-search Engine Construction

Among the first issues to address in meta-search engine construction is database selection. Most of the approaches rank the databases based on the relevancy of the results. For example, GIOSS [6] uses an approach called metaindex to select databases that are likely to contain the desired information. This approach has been proven not flexible or rather ignorant towards the newly added databases. Therefore, re-training is needed; and it is not very effective as the process of re-training is time consuming [7]. This paper's approach is to select several big search engines, like Google, Bing, Yahoo!, etc. The idea is that these search engines index a big part of the web so the likely hood of users getting their desired information is very high. The second issue in designing a meta-search engine is results merging. Traditionally, a linear combination (LC) of score scheme is used to rank the results from different search engines [8]. Although good results are achieved in specific cases, this technique has not yet been shown to produce reliable improvement [9]. MetaCrawler [10] is a popular meta-search engine that employs LC scheme. [11] introduces a meta-search engine called iXmetafind which uses Mearf instead of the traditional LC scheme. It is stated by the authors of [11] that Mearf outperforms LC scheme because it takes advantage of several observations like: presence of the same documents in the results of different search engines in top ranks, common themes, in addition to personalization and clustering methods.

3 *Multi-search* Architecture

As shown in Figure 1, when a user submits a query, the query analyzer first tokenizes the query into uni-gram, bi-gram and tri-gram tokens. Then, it checks whether each of these tokens is defined in the ontologies or not. For those tokens that are defined in the ontologies, semantic networks that represent the query terms and relations between them are constructed. In this context, an ontology may produce zero, one or more semantic networks. Therefore, a merging mechanism is required to merge the produced semantic networks into a single coherent network. This network represents a cooperative decision made by multiple ontologies on the semantics of the query.

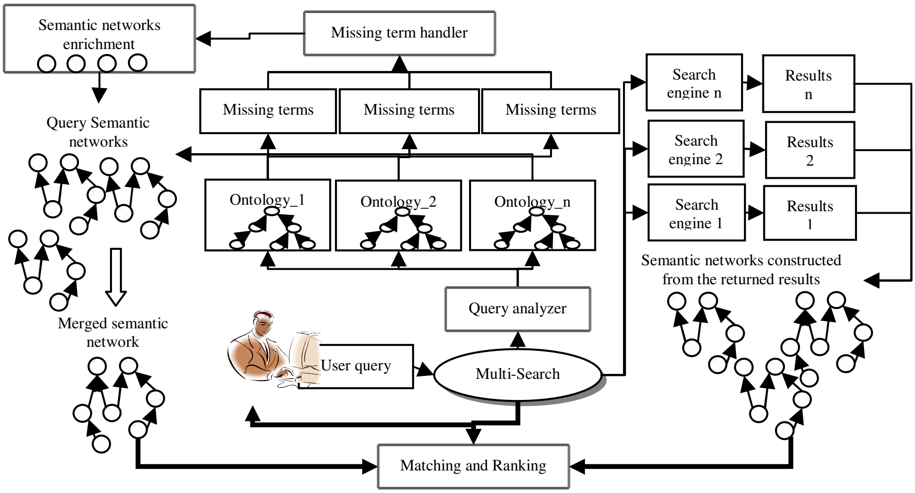


Fig. 1. Architecture of *Multi-Search*

Although using multiple ontologies provides broader domain coverage, we may still have some query terms that are not defined in any of the ontologies. In this case, the missing term handler is utilized to measure the semantic relatedness between the query terms that are missing in the ontologies and those that are defined in them. Based on this technique, a set of additional query terms are suggested to enrich the merged semantic network. On the other hand, *Multi-Search* dispatches the user query into several search engines. At this step, different results may be returned by different search engines. Each returned result is analyzed using the same technique that we used to analyze the user query. As such, semantic networks are constructed from the returned search results. In this context, a returned search result may produce zero, one or more semantic networks. Therefore, *Multi-Search* merges these networks by employing the same technique used to merge the query semantic networks. To filter and rank the results, a scoring function is employed to match the query semantic network and the returned results semantic networks. The higher the similarity, the more it is considered relevant to the query intent. Finally, the user is provided with a decision-oriented mechanism that allows him to contribute in the ranking process.

4 Details of the Proposed Methods in *Multi-Search*

Before we detail the methods of the proposed system, we formalize the use of the terms "Ontology", "Semantic Network", "Semantic Network Merging", "Semantic Network Enrichment", and "Normalized Retrieval Distance (NRD)":

Definition 1: Ontology: An ontology Ω is *quintuple,* $\Omega := \langle C, P, I, V, A\rangle$ where:

- *C* is the set of concepts of the ontology. The concept hierarchy of Ω is a pair $(C, \leq)$, where $\leq$ is an order relation on *C* x *C*. We call $c \in C$ the set of concepts, and $\leq$ the sub-concept relation.
- *P* is the set of properties.
- *I* is the set of instances or individuals
- *V* is the set of property values
- *A* is the set of axioms (such as constraints)

Definition 2: Semantic Network: A semantic network $\zeta := \langle T, R, A\rangle$ where:

- *T* is the set of terms in the network. These terms are query terms that are defined in the ontologies.
- *R* is the set of relations between the query terms. These relations are derived from the ontologies.
- *A* is the set of axioms defined on the query terms and relations.

Definition 3: Semantic Network Merging: A semantic network merging algorithm takes a given set of semantic networks $S= \{\zeta_1, \zeta_2, \zeta_3, \zeta_n\}$ as input and produces a single merged semantic network ζ_{merged} as output.

Definition 4: Semantic Network Enrichment: A semantic network enrichment algorithm takes a given set of query terms that are not defined in the ontology $W= \{w_1, w_2, w_3, w_n\}$ and the merged semantic network ζ_{merged} as input and produces for each $t \in T$ in ζmerged a set of $S(t) \subseteq W$ as output. where,

- *S*(t) is the set of suggested enrichment candidates for t. A suggested candidate $w \in W$ is a word or compound word from *W*.

The set of suggested enrichment candidates *S*(t) can be obtained using the Normalized Retrieval Distance (NRD) function and based on a threshold value *v* using equation 1.

$$S(t,v) := \{\ w \in W \mid \mathrm{NRD}(t,w) \geq v\} \tag{1}$$

Definition 5: Normalized Retrieval Distance (NRD): is a general case of the Normalized Google Distance (NGD) [12] function that measure the semantic relatedness between pairs of terms: Given two terms T_{mis} and T_{in}, the Normalized Retrieval Distance between T_{mis} and T_{in} can be obtained as follows:

$$NRD\ (Tmis\ ,Tin\ ) = \frac{\max\{\log\ f\,(Tmis\ ), \log\ f\,(Tin\ )\} - \log\ f\,(Tmis\ ,Tin\ )}{\log\ M\ - \min\{\log\ f\,(Tmis\ ), \log\ f\,(Tin\ )\}}$$

where,

- T_{mis} is a term that is not defined in the ontology
- T_{in} is a term that exists in the ontology
- $f(T_{mis})$ is the number of hits for the search term T_{mis}
- $f(T_{in})$ is the number of hits for the search term T_{in}
- $f(T_{mis}, T_{in})$ is the number of hits for the search terms T_{mis} and T_{in}
- M is the number of web pages indexed by the search engine

4.1 Query Translation and Returned Search Results Merging

4.1.1 Multiple Ontology-Based Query Translation

First, we apply several Natural Language Processing (NLP) steps on the user query such as stop word removal, n-gram query tokenization, and part-of-speech tagging. After the NLP steps, query tokens are submitted to each of the ontologies to check whether they are defined in them or not. Tokens that are defined in the ontologies are considered as meaningful query terms and thus, semantic networks that represent these terms and relations holding between them are constructed. As a consequence of this step, different number of semantic networks may be produced according to different ontologies. Therefore, we utilize the ontology merging algorithm to merge these networks into a single coherent network. The next example illustrates the NLP steps and the semantic networks construction and merging techniques.

Example1: Query = "Java or jawa the island of Indonesia"

In this example, we use WordNet [13] and OpenCyc [14] ontologies. First, the stop word removal function removes stop words based on a pre-defined list. For example, the words (the, of) are removed from the query. Then, the n-gram tokenization algorithm tokenizes the query into unigram, bigram and trigram tokens. After this step, each token is submitted to each of the ontologies to check whether it is defined in it or not. The algorithm returns that the terms {Java, Island, Indonesia} exist in both WordNet and OpenCyc ontologies. For this set of terms, semantic networks are constructed based on both ontologies as shown in Figure 2.

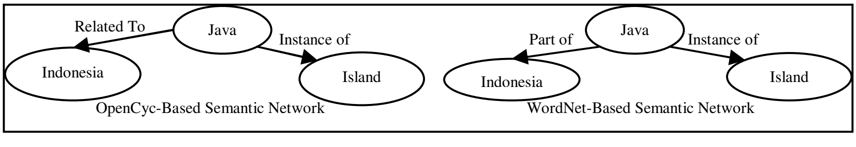


Fig. 2. Query terms represented by semantic networks

As we can see from figure 2, it is not necessarily that the used ontologies produce the same semantic networks. Therefore, due to the semantic heterogeneity between the produced networks, we utilize the merging algorithm described in section 4. In this algorithm, we used the merging techniques proposed in our previous work [15]. The result of merging the semantic networks is shown in figure 3 below.

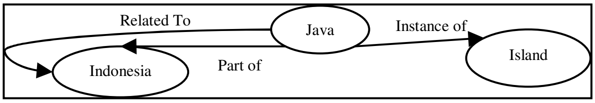


Fig. 3. Merged Query's Semantic Network

The rest of n-gram tokens such as "jawa" are considered as missing background knowledge from the ontologies. However, we don't ignore such tokens because we believe that they may be related to the query terms and can be further suggested to enrich the merged query semantic network.

4.1.2 Statistically-Based Semantic Relatedness Measures

We utilize statistically-based semantic relatedness measures to compensate for the lack of domain coverage in the used ontologies. For query terms that are not defined in the ontologies, we attempt to find whether they can be suggested as candidates to enrich the merged query semantic network. To do this, first we utilize the NRD function described in section 4. This function measures the semantic relatedness between the query terms in the merged query semantic network and other terms that are not defined in any of the used ontologies. As different semantic relatedness measures are returned according to several search engines, we sum up all NRD values for each candidate term. This summation represents a cooperative decision made by several search engines on the semantic relatedness measurers. Table 1 shows the obtained semantic relatedness measures for the term "jawa".

Table 1. Semantic Relatedness Measures for the Term "jawa"

Term \ Term	Java	island	Indonesia
jawa	0.72	0.56	0.69

4.1.3 Semantic Relations Extraction

Obtaining semantic relatedness measures is a prior step towards deriving the actual semantic relation(s) that may hold between semantically related terms. To do this, we defined a list of lexico-syntactic patterns to derive synonymy, hypernymy and hyponymy relations. These types of relations can be automatically obtained by utilizing the Semantic Relation Extractor (SRE) function. For each pair of semantically related terms, the SRE returns the number of their hits by submitting each of the patterns to several search engines. As shown in Algorithm 1 below, for each pattern, the makeQuery function (Line 6) submits exact match queries including both terms. We considered both singular and plural forms of the terms. Patterns that include negation operators such as “No T_*missing* is a(n) T_*in*” are excluded. For instance, to find the relation between the terms “jawa” and “island”, we utilize the SRE function by submitting patterns in the form of exact match queries *Qi* such as, *Q1=“jawa is an island”,* which outputs 80,700 hits result, Q2=*“jawa is a part of island”,* which outputs 0 hits result, and *Q3=“jawa is same as island”,* which outputs 0 hits result.

Algorithm 1. Semantic Relation Extractor Function

```
Input: Semantically related terms,( $T_{\_missing}$, $T_{\_in}$)
Output: suggested relations between terms
String [] suggestedRelations, [] Patterns;
int[] value;
    for each $t_{\_missing} \in T_{\_missing}$
        for each $t_{\_in} \in T_{\_in}$
          for each pattern $_p$ in Patterns
            value.add(makeQuery(“$t_{\_missing}$”, $p$, “$t_{\_in}$”));
          end for
        suggestedRelations.add(max(value));
        end for
end for
```

Based on the number of hits returned for the queries *Qi*, relations defined in the patterns are suggested to be used to enrich the semantic network with the term “jawa”.

4.2 Returned Results Processing

To process the returned results by individual search engines we utilize the query translation techniques explained in section 4.1. First, each returned result page is processed using the NLP steps. At this step, tokens of the result page are matched to the terms in the merged query semantic network. To do this, we employ the Jaro-Winkler distance function [16] which is a simple technique that measures the similarity between the strings of terms in the returned page results and the merged query semantic network. If the similarity measure is above than a threshold value v=0.92, then both strings are considered as equivalent. For example, if we have the term “Object Oriented Programming” in the query semantic network and the term “Object_Oriented Programming” in one of the result pages, then using the string distance function we find that both terms are equivalent. For other terms in the result pages that didn’t match the terms of the merged query semantic network, we utilize the statistical techniques detailed in section 4.1.2. Finally, each returned page result is ranked according to the similarity between the obtained set of its terms and the terms in the query semantic network. Finally, *Multi-Search* provides users with a decision-oriented mechanism that allows them to rank the returned results according to their relevancy to the query intent. In this context, a user can give weights to the returned results and filter out those results that are not semantically related to his query.

5 Experimental Results

This section describes the experiments carried out to evaluate the performance of the proposed meta-search engine. All solutions are implemented in Java and experiments are performed on a PC with dual-core CPU (3000GHz) and 2 GB RAM. The operating system is OpenSuse 11.1. The developed prototype operates on top of big search engines such as (Bing, Google, and Yahoo!). We carried out experiments using WordNet [13], OpenCyc [14], and Yago [17] ontologies. Additional experimentations through a focused study with the help of ten computer science students were carried out to see the significance of the developed system.

5.1 Experiments Using Query Samples

In order to testify our proposal of using multiple ontologies, we selected 45 sample queries (15 per domain) from different domains. We evaluated the precision of the proposed system by comparing human judgments to the automatically returned results when using a single ontology and multiple ontologies. As shown in Table 2, the precision of using multiple ontologies is higher than using a single ontology.

Table 2. Summary of the Obtained Results

Queries per Domain	Precision Using a Single Ontology (WordNet)	Precision Using Multiple Ontologies (WordNet, OpenCyc and Yago)
Countries and Cities	44%	82%
Sports	40%	75%
Programming Languages	42%	71%

5.2 Focused Study Experiments

Phase 1: *Multi-Search* and the search engines that are used in the prototype are disguised so that the users don't not know which one is "*Multi-Search*". Several queries were pre-defined and submitted to these search engines. The interviewees were asked to show which one gives the most relevant results. Among the queries that were given are: "Java Beverage", "Blood Pressure Vital Sign" and "Tree plant".

Phase 2: Given the following scenario, the user has to provide the query:
Let's say you are looking for information about states in the world, you type "state" as your query, but the search engines went off by mixing up your results with "state of mind", "state of health", "state to express", "chemical state", etc. How would you construct a query, so that it will give you, not the list of the states, but specific information about instances of state? The feedbacks from the 10 students were closely considered. Feedbacks were then compiled together to reflect how does the prototype perform. After the interviews, every interviewee had to answer a set of questions:

1. From scale 1 to 5 (1 = Worst, 5 = Best), how would you rate the results relevancy?
2. From scale 1 to 5 (1 = Worst, 5 = Best), how precise it defines the query semantics?
3. From scale 1 to 5 (1 = Worst, 5 = Best), how close it is in defining ambiguous term?
4. How many average links you had to click before you found the desired result?

5.3 Evaluation of the Study

The study results are compiled and shown in Figure 4. For overall performance, the average rate that was given by the users is= 3.7**.** As it is previously discussed in section

1, among the objectives of *Multi-Search* is to involve users in ranking the returned results. However, some people were confused by the way queries are converted into semantic networks. One argument was that, in most cases, users think of search by submitting keywords instead of actual query semantics.

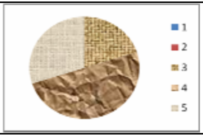
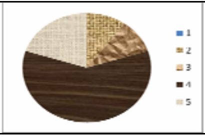
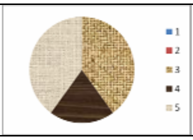
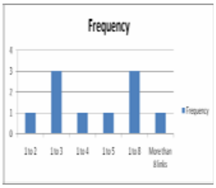
Users' opinions of result relevancy (From scale 1 – 5)	Users' opinions of the semantic precision (From scale 1 – 5)	Precision of disambiguating ambiguous terms	Average links that users clicked to get the desired results
			
Out of 10 people, 2 people rated 3, 5 people rated 4, and 3 people rated 5. Average = 4.1	Out of 10 people, 1 person rated 2, 1 person rated 3, 6 people rated 4, 2 people rated 5. Average = 3.9	Out of 10 people, 4 people rated 3, 2 people rated 4, 4 people rated 2. Average = 4	

Fig. 4. Evaluation of the Study Results

Table 3. Phase 1 Results

Search engine **Query**	Bing	Yahoo!	Google	*Multi-Search*
First Query	4	1	2	3
Second Query	3	2	4	2
Third Query	5	0	1	4
Total	12	3	7	9

From the table above, we can see that Bing and *Multi-Search* are favored among the other search engines. Although at this phase of experiments Bing was given high priority, *Multi-Search* is distinguished by the way it involves users in the search process as it provides them with a mechanism for filtering and ranking the results.

Phase 2 Results: To narrow down the results, the interviewee usually added a word before and/or after the term "state". For example: "State Country", "Country State", "State of 'INSTANCE_NAME'" like "State of Malaysia" and "State of Mississippi", etc. For this scenario, the students were not expecting the search engine to understand the meanings of the queries, but they were expecting that keywords of the queries would be exactly matched to their equivalent keywords in Web pages. The reason is because most conventional search engines search the web based on keywords matching. This approach has influenced the way people search for information over the internet. For instance, when searching for something, some people will put all the keywords that they think will appear in a web page. However, as *Multi-Search* combines ontology-based query translation and semantic relatedness measures, it was able to better understand queries by filling the "semantic-gap" between the meanings of keyword used to index WebPages and keywords used by the user. Therefore, in this case, *Multi-Search* was favored among the other search engines.

6 Conclusion and Future Work

In this paper, we proposed *Multi-Search*, a meta-search engine that employs knowledge represented by multiple ontologies and combines semantic and statistical based techniques to derive the semantic aspects of both the user query and the returned search results. Based on these semantic aspects, relevancy rates are given to the user so that he can filter and rank the results. In the developed prototype experimentations were done with the help of 10 computer science students and evaluations were carried out based on the experimental results. Students agree that the prototype has successfully considered semantics of the query instead of matching keywords. Some of the feedbacks that were given stated that the search engine is very useful for people who are new to a certain topic and would like to find legit information about that topic. Among the future works that we plan to do is to exploit additional ontologies to ensure broader domain coverage and more precise query translation. In addition, instead of manually defining the lexico-syntactic patterns, we plan to use automatic pattern acquisition techniques. The benefits of using these techniques are (i) saving the time and effort required to manually define the patterns and (ii) acquiring additional relations other than synonymy and hyponymy.

References

1. Tanaka, K., et al.: Improving Search and Information Creditability Analysis from Interaction between Web1.0 and Web 2.0 Content. Journal of Software 5, 154–159 (2010)
2. Gulli, A., Signorini, A.: The indexable web is more than 11.5 billion pages. In: The 14th International World Wide Web Conference (WWW), pp. 902–903 (2005)
3. Guarino, N., Masolo, C., Vetere, G.: OntoSeek: Content-Based Access to the Web. IEEE Intelligent Systems 14(3), 70–80 (1999)
4. Gauch, S., Chafee, J., Pretschner, A.: Ontology-based personalized search and browsing. In: Web Intelligence and Agent Systems, pp. 219–234 (2003)
5. Wimalasuriya, D., Dou, D.: Using Multiple Ontologies in Information Extraction. In: CIKM 2009, Hong Kong, China, pp. 235–244 (2009)
6. Gravano, L., Garcia-Molina, H.: Generalizing GlOSS to Vector-Space Databases and Broker Hierarchies. In: Proc. of the 21st VLDB Conference, Switzerland, pp. 78–89 (1995)
7. Tseng, J., Hwang, G.J.: A Study of Metaindex Mechanism for Selecting and Ranking Remote Search Engines. Journal of Computer Science and Engineering, 353–369 (2007)
8. Tang, J., Du, Y.J., Wang, K.L.: Design and Implementation of Personalized Meta-Search Engine based on FCA. In: Proceedings of the Sixth International Conference on Machine Learning and Cybernetics, Hong Kong, China, pp. 4026–4031 (2007)
9. Aslam, J., Montague, M.: Models for Metasearch*. In: Proc. of the 24th Annual International ACM SIGIR Conf. on Research and Development in IR, USA, pp. 276–284 (2001)
10. MetaCrawler (2010), `http://www.metacrawler.com`
11. Han, S., Karypis, G.: Intelligent Metasearch Engine for Knowledge Management. In: Proc. of the CIKM 2003, pp. 492–495 (2003)
12. Cilibrasi, R., Vitanyi, P.: The Google Similarity Distance. IEEE Transactions on knowledge and data engineering 19(3), 370–383 (2007)
13. Miller, G.A.: WordNet: A Lexical Database for English. Communications of the ACM, 409–409 (1995)

14. Matuszek, C., Cabral, J., Witbrock, M., DeOliveira, J.: An Introduction to the Syntax and Content of Cyc. In: AAAI Spring Symposium on Formalizing and Compiling Background Knowledge and Its Applications to Knowledge Representation and Question Answering, Stanford, CA, pp. 44–49 (2006)
15. Maree, M., Belkhatir, M.: A Coupled Statistical/Semantic Framework for Merging Heterogeneous domain-Specific Ontologies. In: Accepted for Publication in the Proceedings of the 22th International Conference on Tools with Artificial Intelligence, France (2010)
16. Winkler, W.E.: The State of Record Linkage and Current Research Problems. Publication R99/04, Statistics of Income Division, Internal Revenue Service (1999), http://www.census.gov/srd/www/byname.html
17. Fabian, M.S., Gjergji, K., Gerhard, W.: YAGO: A Core of Semantic Knowledge Unifying WordNet and Wikipedia. In: 16th International World Wide Web Conference, pp. 697–706 (2007)