3M1-IOS-3a-1

HOMME: Hierarchical-Ontological Mind Map Explorer

Yi-Shin Chen, Pei-Ling Hsu, Hsiao-Shan Hsieh, Li-Chin Lee, Carlos Argueta

Institute Information Systems and Applications,
National Tsing Hua University

In this paper we present HOMME (Hierarchical-Ontological Mind Map Explorer), a novel application expanding the way users explore the web. HOMME offers the user two powerful tools. *Ontology builder* creates a mind map around the input query, presenting the underlying semantic relationships between the query and other terms on the web. *Concept Linker* provides the opportunity to experience the crossing of the boundaries of conceptual spaces. Our demonstration shows the ability of HOMME to convey significant knowledge to the user. Furthermore, future research on Web Information Integration will be possible by leveraging the power of the ontological knowledge provided by HOMME.

1. Introduction

Ever since ancient times, collecting all human knowledge has been a recurring goal. The advent of the computer, followed by the Internet and the World Wide Web, have made this goal a reality. The Web has become the largest repository of human knowledge in the world. As open as the Web is, it allows for anyone with Internet access to generate knowledge. Coming from different sources, this knowledge is often scattered, mutually redundant, and mutually complementary. It is crucial, then, to find ways to integrate all this knowledge.

Integration has been considered by researchers. They have manually [5, 18, 25], semi-automatically [27, 6, 3, 22, 26], or automatically organized web data based on some expert (or crowd-generated) knowledge.

Among the crowd-generated knowledge (also termed crowd wisdom), Wikipedia is the most famous and successful example. It has been investigated and utilized for organizing web data by a number of research projects [24, 2]. Wikipedia's data can be classified as unstructured articles, plus structured information (such as infoboxes and classication) where the structured data are frequently used as a standard for integrating web information. YAGO [24] and DBpedia [2]] are two famous systems that utilize infoboxes and classication data to construct a Web taxonomy and semantic relationships, respectively. A Web taxonomy can help users to navigate web data *hierarchically*, and semantic relationships can help users to explore web data *horizontally*.

As the success of the organizing the data based on Wikipedia information, other kinds of crowd wisdom, such as social annotations and search logs, have also been studied by researchers [15, 16, 23, 17, 1, 8, 7, 9, 21, 19]. However, these papers faced the challenges of seeking integrating standards. Two approaches to address this are proposed. The first approach is relying on external resources [15, 16, 23], such as WordNet or Yago, and mapping the terminologies between the external resources and the integrating data. This approach can construct a taxonomy and semantic relationships. However, the proportions of successful mapped terminologies across different data resources are relatively low.

The second approach is directly constructing a taxonomy and relationships from the bottom-up [17, 1, 8, 7, 9, 21, 19]. In this approach, the data are represented by graphs, sets, or statistical matrixes. Afterward, some graphs, sets, or matrix algorithms are applied to find a taxonomy and relationships. However, since the manually generated standards are not employed in these processes, this bottom-up approach usually has difficulties in labeling the relationships semantically.

In this paper, we assembled and leveraged the notions of the previous two approaches. We present HOMME*1 (Hierarchical-Ontological Mind Map Explorer), a mind mapping tool that can create up-to-date ontological representations of the knowledge contained in the Web. Multiple heterogeneous data resources with crowd wisdom are utilized in HOMME.

Rather than relying mainly on structure data as the integrating standards, HOMME formally models knowledge by normalizing the heterogeneous data and applying algorithms to extract the semantic relationships. By taking the normalized data as anchors, the relationships can be constructed from the bottom up. The HOMME system can help users to navigate the conceptual relationships between terms horizontally. In addition, it also provides horizontal views to explore concept domains of the terms. HOMME presents the possibility of automatically constructing a hierarchical-ontological mind map from unstructured and heterogeneous data resources, which is a supplementary knowledge from that constructed from Wikipedia.

An overview of HOMME is described in Section 2. Section 3. describes the integrator for the heterogeneous data. Section 4. describes the reasoning behind various semantic relationships. Section 4.2 describes clustering to construct hierarchical structures. The experiment is reported in Section 5. Section 6. concludes the paper and discusses future work.

*1 Homme means the human being in French, and so we have used this name for our system, which contains human intelligence.

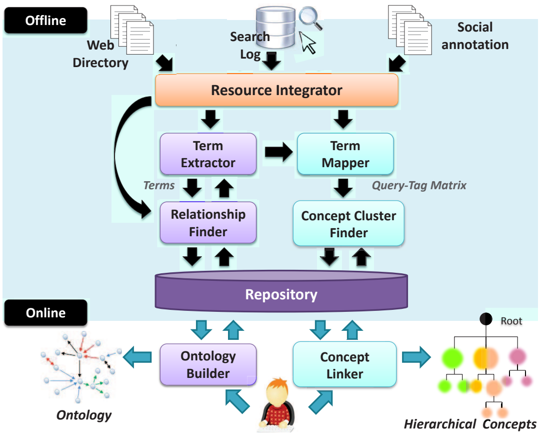

Figure 1: Framework

2. Overview

HOMME contains multiple data resources, including search logs and social annotations, data from the Open Directory Project(ODP)[*2], and Delicious[*3] tags. These heterogeneous data have *different forms*, *ambiguous terms*, and *scattered information*. These are all involved with the issues of form. For example, concepts are represented in dierent lexical formats; terms in the same form likely refer to different concepts; and scattered information means that content in various lexical formats refer to the same concepts that are in dierent web pages.

To address the issues of form, two components are proposed (illustrated in Figure 1). One is the *Relationship Finder*, which constructs relationships among terms and web data. The other is the *Concept Cluster Finder*, which identifies different aspects (or concepts) of terms to clarify meaning.

The framework of this work is illustrated in Figure 1. During the offline process, the three data sources are normalized and integrated by the *Resource Integrator*. These normalizations are formally defined in Section 3.

Both the *Term Extractor* and *Term Mapper* utilize the previously normalized data set as input. For each given concept c, the *Term Extractor* looks through the whole data set for the terms related to c and generates a corresponding list as the output. The *Term Mapper* then employs the output list to build a Query-Tag Matrix by first classifying terms according to the categories in ODP and using the frequency of the tags assigned to each query as features.

Once the terms list and the Query-Tag Matrix are ready the *Relationship Finder* and *Concept Cluster Finder* modules are activated. Section 4. and Section 4.2 describe them in detail.

3. Resource Integrator

In order to integrate the heterogeneous data, the data are normalized and decomposed into the smallest elements, which share common characteristics.

Definition 1 (Word Sequences): We let $\langle\mathbb{W}\rangle$ be the set of all of the words. A finite word sequence ws with length m can also be represented as $\langle w_1, w_2, \cdots, w_m\rangle$, $w_i \in \mathbb{W}$. We consider a word sequence with only one word is the same as the word, i.e., for all $w \in \mathbb{W}, \langle w\rangle = w$, $\langle\mathbb{W}\rangle$ is the set of all finite word sequences.

Definition 2 (Concept Sequences): A sequence of words can represent concept; we define a notation, $\mathbb{C} = \langle\mathbb{W}\rangle$. A finite concept sequence with length n can be represented as $\langle c_1, c_2, \cdots, c_n\rangle$, $c_i \in \mathbb{C}$. $\langle\mathbb{C}\rangle$ is the set of all finite concept sequence. According to Definition 1 and Definition 2, we has the following properties: $\mathbb{W} \subset \mathbb{C}$, $\langle\mathbb{W}\rangle \subset \langle\mathbb{C}\rangle$.

Based on these definitions, several operations and extractions of word sequences are defined formally as follows:

Definition 3 (Concept Operators): Let $a, b \in \langle\mathbb{C}\rangle$ be two concept sequences, $a = \langle a_1, a_2, \cdots, a_{n_a}\rangle$, $b = \langle b_1, b_2, \cdots, b_{n_b}\rangle$,

- a_i denote the i^{th} concept in a sequence
- $|a| = n_a$ number of concepts in this sequence
- $gram_k(a) = \{\langle a_i, a_{i+1}, \cdots, a_{i+(k-1)}\rangle | i \in \mathbb{N} \cap [1, n_a - (k-1)]\}$ the k-gram of a
- $strip_k(a) = \langle a_1, a_2, \cdots, a_{n_a-k}\rangle$ a concept sequence with last k concepts striped

After these definitions, the data are integrated as follows:

Definition 4 (Query): We let $\mathbb{K}_c = \bigcup\{(qid, time)\}$ a primary key set of click log data, $q_c(k_c) : \mathbb{K}_c \longrightarrow \langle\mathbb{W}\rangle$ is the query of k_c, for k_c in $\mathbb{K}_c$. Therefore, the set of all queries are extracted and defined as: $Query_c = \bigcup\limits_{k \in \mathbb{K}_c} q_c(k_c)$.

Definition 5 (URL): Let $\mathbb{L}$ be the set of all URLs. Each URL $l \in \mathbb{L}$ is departed into hostname $h(l) : \mathbb{L} \longrightarrow \mathbb{W}$, resource path $p(l) : \mathbb{L} \longrightarrow \langle\mathbb{C}\rangle$. In addition, let $\mathbb{H}$ be the set of all hostnames. A hostname, $h \in \mathbb{H}$, can be further decomposed as domain name $d(h) : \mathbb{H} \longrightarrow \langle\mathbb{C}\rangle$, second level domain $d_2(h) : \mathbb{H} \longrightarrow \langle\mathbb{W}\rangle$, and third level domain $d_3(h) : \mathbb{H} \longrightarrow \langle\mathbb{W}\rangle$

Definition 6 (Social Annotations): Delicious tag of an URL, l, can be represented as $\alpha(l) : \mathbb{L} \longrightarrow \langle\mathbb{W}\rangle$. ODP data contain ODP title $\tau_o(l) : \mathbb{L} \longrightarrow \langle\mathbb{W}\rangle$ and ODP category $c_o(l) : \mathbb{L} \longrightarrow \langle\mathbb{C}\rangle$. In addition, ODP categories are defined as set $Category_{ODP} = \bigcup\limits_{u \in Url} c_o(u)$, and each of them is decomposed as sub-category $c_{o,i}$ Each ODP category contains sub-categories, $i \in \mathbb{N}$.

*2 http://dmoz.org/
*3 http://delicious.com

4. Relationship and Concept Cluster Finder

Two different finders are described in this section. The *Relationship Finder* which seeks to find important semantic relationships between word sequences, such as "*ae*" has-Meaning "*American eagle*" and "*ae*" has-Website "*www.ae.com*," is described in Section 4.1. The *Concept Cluster Finder*, whose ultimate objective is to organize the related terms hierarchically based on their concepts, is described in Section 4.2.

4.1 Relationship Finder

During the processes of Relationship Finder, word sequences are like seeds, for the following reasons. First, word sequences are often composed of several atomic terms, which can link the different data sets together. Second, the meaning and intentions of the word sequences are often ambiguous. For example, multiple query terms might represent: (1) one compound name entity; (2) several concepts; or (3) one primary concept with several modifiers.

Since word sequences are the basic components of many heterogeneous data sets, the Relationship Finder can also work for those data sources whose data can be modeled as word sequences. Several deduction rules are introduced:

- **Has-Subclass Relationship Finder**: After observing word sequences from queries and URLs, we found that successive terms among word sequences are likely to be hierarchically related, but the order of the words in a word sequence cannot guarantee the direction of specialization. For example, compound terms can be the subclass of the modified term, such as "Law School" is a subclass of "School." For another example, the subsequent noun can be the subclass of the prefix, such as "Travel Agent" is a subclass of "Travel." Therefore, this finder attempts to match up the word sequences and ODP categories. Once the match exists frequently, a "has-subclass" relationship is then established (e.g. "travel" Has-Subclass "travel agents"). The rule of Has-Subclass is defined as follows:

  $\mathcal{R}_{HSC} = \big\{(a, \langle a, b\rangle)\big|\langle a, b\rangle \in \bigcup_{c \in Category_{ODP}} gram_2(c) \cap \langle \mathbb{W}\rangle\big\}$

- **Has-Data-About Relationship Finder**: Similar to the previous relationship finder, we observed that terms in word sequences likely denote sub-concepts among web pages. This finder tries to find a match between the query terms and the resource path of the clicked URLs. Once there is a frequent match, the finder constructs a successive "has-Data-About" relationship from the domain name to the final terms of the resource path which matched the submitted queries. For instance, from the URL www.mtv.com/music/artist/bowlingforsoupartist.jhtml, clicked after submitting the query "bowling for soup," we can determine that "mtv" has-Data-About "music" has-Data-About "artist" has-Data-About "bowling for soup."

  The rule of Has-Data-About is defined formally as follows:

  $\mathcal{R}_{HDA} = \bigcup_{k,u}\Big\{(a,b)\Big|\langle a,b\rangle \in gram_2\Big(\langle h(l)(u)\rangle + strip \circ rp(u))\Big)\Big\}$ where $k \in \mathbb{K}, u = l_c(k)$

- **Has-Website Relationship Finder**: Has-Website relationships gather web sites and their concepts in various lexical formats. This finder matches the terms of the query with those of the URL's hostname. When the match occurs frequently, this relationship is constructed by the following rule:

  $\mathcal{R}_{HWS} = \bigcup_{u,c}\{(c,h)\big|h = h(u)\}$ where $u \in \mathbb{L}$, c is a frequent matched word sequence

- **Has-Meaning Relationship Finder**: Terms in word sequences likely refer to the same concepts, but in different forms. For example, the queries of the word sequences "ae," "American Eagle," and "American Eagle Outfitter" refer to the same concept, the web page "www.ae.com." This finder tries to find distinct queries and ODP data referring to the same concepts.

  To achieve this goal, two procedures are conducted. First, queries referring to the same concepts are grouped. Grouping queries is a challenging research issue, since the concepts behind the queries are usually uncertain in nature. However, based on previous research studies [4, 20, 11, 12, 14, 10], queries can be grouped in a more straightforward approach; that is, based on their *navigational intention*. These studies have suggested that the intentions of certain popular queries can be directly inferred from the clicked URLs, especially when the number of average clicked URLs is less than two [13]. Inspired by this, the finder groups navigational queries based on the clicked URLs.

  Second, social annotation data are integrated with group queries based on their referring URLs. Since Delicious data might consist of numerous annotated tags, the integrating process might be too complicated to finish. To simplify the issue and to demonstrate that this relationship finder is valuable, we have only adapted the ODP data in this paper. The rules for this finder are formally defined as follows:

  $\mathcal{N}$: navigational query Let $\mathbb{K}_c|_{\mathcal{N}} = \big\{k\big|k \in \mathbb{K}_c, query_c(k) \in \mathcal{N}\big\}$

  For all $k \in \mathbb{K}_c|_{\mathcal{N}}$
  Let $q = q_c(k_c), ot = \tau_o(l) \circ l_c(k_c), sld = d_2(h) \circ url_c(k)), h = h(l) \circ l_c(k_c)$

  $\mathcal{R}_{HMN} = \big\{(q, ot)\big|q \subset ot \vee q = abbr(ot)\big\}$
  $\cup \big\{(sld, ot)\big|abbr(ot) = sld\big\}$
  $\cup \big\{(q, h)\big|ot = \langle\rangle, q \subset sld\big\}$

The newly discovered relationships are then used to improve the term extraction process in an iterative interaction between the *Term Extractor* and the *Relationship Finder* modules. After numerous iterations, the offline processes can generate a huge ontology containing all of the relationships between the corresponding terms.

This output can be benecial for the information integration techniques, but are too large for human beings to digest. Therefore, in the online demonstration, the user only obtains a small part of the graph when submitting a query. When a query is submitted, the Ontology Builder module nds all the related terms that hold relationships with the target query in the repository. It then creates the corresponding graphical representation, showing the terms and annotated links between them.

4.2 Concept Cluster Finder

Concept Clustering uses the Query-Tag Matrix generated by the *Term Mapper* as the input of the clustering process. Note that the size of the Query-Tag Matrix is extremely huge*4.

This module employs the k-means algorithm to generate the initial clusters, which could be split or merged in the later stages. The splitting decisions are made based on the intra-distance between cluster centroids and queries. The merging decisions come from the analysis of the inter-distance between clusters. These refinement processes are performed iteratively until none of the clusters meet the splitting/merging criteria. Finally, each cluster will be labeled automatically, based on the representative of the cluster. In this demonstration, the cluster labels are the features with the top score.

5. Experimental Evaluation

We developed a prototype HOMME system based on the proposed approaches in Section 4. Human experts sampled some of the outputs of the HOMME system and veried their accuracy. The setup of the prototype system, sampling outputs, and the experimental results are presented in Section 5.1, Section 5.2, and Section 5.3.

5.1 Setup

We gathered three data resources: search logs, ODP data, and Delicious tags. The search logs were released by Microsoft Live Labs, and where collected from a sample of United States users in May 2006. The ODP data are open for users to download its structure and content data. The Delicious tag were crawled from February 2010 to May 2010. In addition, 1,512,556 navigational queries were extracted according to the approach that was proposed in the previous research project [13].

The data were preprocessed, for example, by removing symbols in the data resources. The queries were input as word sequences in the Has-Subclass, Has-Data-About, and Has-WebSite Relationship Finders. The interaction between the queries and the clicked URLs was considered and observed in order to determine the frequent matches in the three relationships. For example, matches were considered

*4 In our system, each category contains more than tens of thousands tags.

Figure 2: Mind Map for the query “apple”

Figure 3: Aspects for the query “apple”

frequent only when the queries were frequent enough, which we decided was more than five times, to reduce bias. The correlated distributions between these frequent queries and the clicked URLs were calculated to determine the frequent matches between the queries and the clicked URLs.

The prototype system was implemented in PHP and JavaScript InfoVis Toolkit on Windows.

5.2 Demonstration

Examples of the outputs of the HOMME systems are displayed in two figures: Figure 2 and Figure 3.

Figure 2 illustrates the mind map of the *Ontology Builder*. This mind map shows the different lexical formats for the same concepts. For example, “apple” and “apple computer” can refer to the same concept. It also indicates the relevant web resources for the query “apple.” For example, the “apple company,” whose official website is *www.apple.com*. This official website contains information about “*apple_itunes*” and “*apple quicktime*.”

Figure 3 displays the concept spaces of the *Concept Linker* for the query “apple.” The query “apple” is the core of this graph. Its various direct aspects are shown as orange

Figure 4: Aspect *Computer* of the query "apple"

nodes. Each aspect has sub-aspects in green nodes. When the orange nodes were observed, these aspects of "apple" were more than those guessed by individual users. This suggests that crowd wisdom does extend more aspects of specific words than individuals provide. To satisfy the curiosity of individual users and give them further insight into the extended aspects of a term, each node of the concept spaces can be clicked on.

Figure 4 is an example of when the aspect *Computer* is clicked. A list of relevant queries is displayed.

These queries are associated with both the direct parent node, "apple," and the current aspect, *Computer*. While Figure 4 contains a long list of such associated queries, other aspects usually contain one to three queries. This leads to more sub-aspects of *Computer* than those of the others.

5.3 Experimental Results

Human experts sampled some of the outputs of the HOMME system to determine whether the relationships and aspects were correct. While evaluating the accuracy of the aspects, we compared our method with two different clustering methods. One was single-link agglomerative hierarchical clustering(AHC), which is bottom-up hierarchical clustering. Another was DBSCAN, which is density-based clustering that can discover clusters with arbitrary shapes. Figure 5 shows the results of our comparison with AHC. Our method was much better than AHC in Sport and Science, while AHC was better than ours in Health. Figure 6 privdes the results of our comparison with DBSCAN. Except for in the Health category, our method performed better than DBSCAN.

The accuracy of the relationships was also evaluated by human experts. The results are shown in Figure 7. This figure reveals several suggestions. First, the accuracy of Has-Subclass was the lowest among all relationships. A possible reason for this could be that while users determined this relationship, they considered the properties of inheritance in Has-Subclass. However, the design of this relationship finder has not been guaranteed to contain the properties of inheritance. Second, the relationships Has-Subclass and

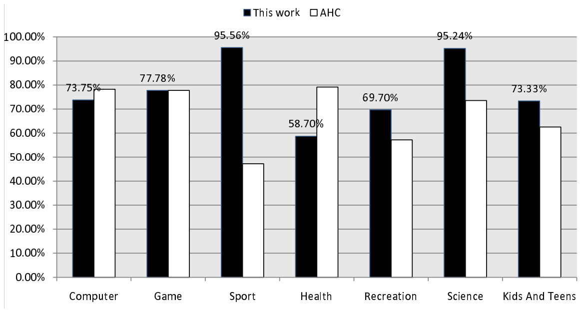

Figure 5: Accuracy comparison with AHC

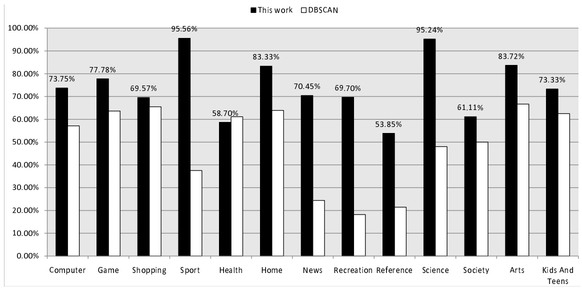

Figure 6: Accuracy comparison with DBSCAN

Has-Website achieved high accuracy. This suggests that the relationships with "frequent match" likely represent opinions which are popular enough to be acceptable. Third, Has-Meaning also reached high accuracy. Perhaps this was because the condition for lexical formats for this relationship finder is strict. Generally speaking, the accuracy of the all relationships was high. This suggests that these detected relationships are recognized by humans.

Figure 7: Accuracy comparison with DBSCAN

6. Conclusion and Future Work

In this paper, we proposed our HOMME systems. The general idea and the HOMME framework were described. Two approaches for constructing the two main components of this system were proposed, executed, and evaluated. The results of the evaluations were highly accurate.

The executions and the evaluations suggest that the proposed approaches can cluster aspects of a concept and construct semantic relationships automatically. Moreover, they are enable the constructed results to be recognized by humans. However, to improve our approaches further, we could work on issues of efficiency, such as: (1) the coverage rates between the constructed results and seed data resources, (2) execution times, (3) extensions of various types of relationships, and (4) the impact of each limitations on each relationships finders.

References

[1] R. Baeza-Yates and A. Tiberi. Extracting semantic relations from query logs. In *Proceedings of the 13th*

ACM SIGKDD international conference on Knowledge discovery and data mining, pages 76–85, 2007.

[2] C. Becker and C. Bizer. Exploring the geospatial semantic web with dbpedia mobile. *Web Semantics: Science, Services and Agents on the World Wide Web*, 7(4):278–286, 2009.

[3] C. Bizer, J. Lehmann, G. Kobilarov, S. Auer, C. Becker, R. Cyganiak, and S. Hellmann. Dbpedia - a crystallization point for the web of data. *Web Semant.*, 7:154–165, September 2009.

[4] A. Broder. A taxonomy of web search. *SIGIR Forum*, 36(2):3–10, 2002.

[5] A. Farquhar, R. Fikes, and J. Rice. The ontolingua server: a tool for collaborative ontology construction. Technical report, 1996.

[6] B. Fortuna, M. Grobelnik, and D. Mladenic. Ontogen: Semi-automatic ontology editor. In *Proceddings of Human Interface and the Management of Information. Interacting in Information Environments, Symposium on Human Interface*, volume 4558, pages 309–318, 2007.

[7] A. P. Francisco, R. Baeza-Yates, and A. L. Oliveira. Mining large query induced graphs towards a hierarchical query folksonomy. In *String Processing and Information Retrieval*, volume 6393, pages 238–243, 2010.

[8] A. P. Francisco, R. Baeza-Yates, and A. L. Oliveira. Mining query logs induced graphs. In *Proceedings of Tech. Rep. INESC-ID*, volume 36, 2010.

[9] A. P. Francisco, R. Baeza-Yates, and A. L. Oliveira. Mining query log graphs towards a query folksonomy. *Concurrency and Computation: Practice and Experience*, 2011.

[10] B. J. Jansen, D. L. Booth, and A. Spink. Determining the informational, navigational, and transactional intent of web queries. *Inf. Process. Manage.*, 44(3):1251–1266, 2008.

[11] I.-H. Kang and G.-C. Kim. Query type classification for web document retrieval. In *SIGIR*, pages 64–71, 2003.

[12] U. Lee, Z. Liu, and J. Cho. Automatic identification of user goals in web search. In *WWW '05: Proceedings of the 14th international conference on World Wide Web*, pages 391–400, New York, NY, USA, 2005. ACM Press.

[13] P.-C. Liu, Y.-S. Chen, and K. C.-C. Chang. Are we searching for the same thing? a large-scale analysis of search engine logs. Technical report, Department of Computer Science, National Tsing Hua University, 2007.

[14] Y. Liu, M. Zhang, L. Ru, and S. Ma. Automatic query type identification based on click through information. In *AIRS*, pages 593–600, 2006.

[15] A. Marchetti, M. Tesconi, F. Ronzano, M. Rosella, and S. Minutoli. Semkey: A semantic collaborative tagging system. In *Proceedings of WWW 2007 Workshop on Tagging and Metadata for Social Information Organization*, 2007.

[16] M. Maree, S. M. Alhashmi, and M. Belkhatir. Querysem: Deriving query semantics based on multiple ontologies. In *Web-Age Information Management - 12th International Conference*, pages 157–168, 2011.

[17] P. Mika. Ontologies are us: A unified model of social networks and semantics. *Web Semantics: Science, Services and Agents on the World Wide Web*, 5(1):5–15, 2007.

[18] N. F. Noy, M. Sintek, S. Decker, M. Crubezy, R. W. Fergerson, and M. A. Musen. Creating semantic web contents with protege-2000. *IEEE Intelligent Systems*, 16(2):60–71, 2001.

[19] A. Plangprasopchok and K. Lerman. Constructing folksonomies from user-specified relations on flickr. In *Proceedings of the 13th international conference on World Wide Web*, pages 781–790, 2009.

[20] D. E. Rose and D. Levinson. Understanding user goals in web search. In *WWW '04: Proceedings of the 13th international conference on World Wide Web*, pages 13–19, New York, NY, USA, 2004. ACM Press.

[21] D. Shen, M. Qin, W. Chen, Q. Yang, and Z. Chen. Mining web query hierarchies from clickthrough data. In *Proceedings of the 22nd national conference on Artificial intelligence*, volume 1, pages 341–346, 2007.

[22] L. N. D. Silva and L. Jayaratne. Semi-automatic extraction and modeling of ontologies using wikipedia xml corpus. In *Applications of Digital Information and Web Technologies*, pages 446–451, 2009.

[23] L. Specia and E. Motta. Integrating folksonomies with the semantic web. In *Proceedings of the European Semantic Web Conference*, pages 624–639, 2007.

[24] F. M. Suchanek, G. Kasneci, and G. Weikum. Yago: A core of semantic knowledge. In *16th international World Wide Web conference*, pages 697–706, 2007.

[25] Y. Sure, M. Erdmann, J. Angele, S. Staab, R. Studer, and D. Wenke. Ontoedit: Collaborative ontology development for the semantic web. In *Proceedings of First International Semantic Web Conference*, volume 2342, pages 221–235, 2002.

[26] C. C. Xavier and V. L. S. de Lima. A semi-automatic method for domain ontology extraction from portuguese language wikipedia's categories. In *Advances in Artificial Intelligence*, pages 11–20, 2010.

[27] L. Zhou, Q. E. Booker, and D. Zhang. Toward rapid ontology development for underdeveloped domains. In *Proceedings of the 35th Annual Hawaii International Conference on System Sciences*, page 106, 2002.