REGULAR PAPER

User-centric query refinement and processing using granularity-based strategies

**Yi Zeng · Ning Zhong · Yan Wang ·
Yulin Qin · Zhisheng Huang · Haiyan Zhou ·
Yiyu Yao · Frank van Harmelen**

Abstract Under the context of large-scale scientific literatures, this paper provides a user-centric approach for refining and processing incomplete or vague query based on cognitive- and granularity-based strategies. From the viewpoints of user interests retention and granular information processing, we examine various strategies for user-centric unification of search and reasoning. Inspired by the basic level for human problem-solving in cognitive science, we refine a query based on retained user interests. We bring the multi-level, multi-perspective strategies from human problem-solving to large-scale search and reasoning. The power/ exponential law-based interests retention modeling, network statistics–based data selection, and ontology-supervised hierarchical reasoning are developed to implement these strategies. As an illustration, we investigate some case studies based on a large-scale scientific literature dataset, DBLP. The experimental results show that the proposed strategies are potentially effective.

Y. Zeng · N. Zhong · Y. Wang · Y. Qin · H. Zhou · Y. Yao
International WIC Institute, Beijing University of Technology, 100124 Beijing, China

N. Zhong (✉)
Department of Life Science and Informatics, Maebashi Institute of Technology,
Maebashi 371-0816, Japan
e-mail: zhongn@bjut.edu.cn; zhong@maebashi-it.ac.jp

Y. Qin
Department of Psychology, Carnegie Mellon University, Pittsburgh, PA 15213, USA

Z. Huang · F. van Harmelen
Department of Computer Science, Vrije Universiteit Amsterdam,
1081 HV, Amsterdam, The Netherlands

Y. Yao
Department of Computer Science, University of Regina, Regina, SK S4S 0A2, Canada

1 Introduction

As two important machine problem-solving methods, search and reasoning meets many
barriers when the data goes to large scale. Two of the major challenges are:

1. It is not easy to acquire most relevant data by search, since the search results space may
   be very huge [7].
2. Traditional reasoning systems cannot handle large-scale data in a rational time, which is
   acceptable to users [15].

For the first challenge, refining the search process by reasoning is proposed [7]. The general
idea is that logic can help to deduce whether the results from an initial search are useful [7].
In the book entitled "Weaving the Web", the author argues that "A simple search would have
returned an endless list of possible answers that the human would have to wade through.
By adding logic, we get back a correct answer" [7]. For the second challenge, refining the
reasoning process by search is proposed [15]. The idea starts from the unification of search and
reasoning, which emphasizes on making an intelligent search of subdataset and do reasoning
based on the selected dataset in a reasonable time with user involvement [15,16]. Although
these proposed frameworks are potentially effective, more efforts are needed to implement
them as concrete strategies. We here discuss two further issues:

– The first issue can be summarized as follows. For searching on large-scale data, due to the
  lack of query skills or related background, a user may submit an incomplete or a vague
  query, which may require more processing time to obtain a satisfying answer if the query
  process starts from the query input by the users. Since incomplete or vague queries may
  produce too many irrelevant results, especially under the context of large-scale data. The
  following questions then arise:

  – Should the search process always start from the point of query input made by the user,
    especially when the input is incomplete or vague?
  – As a possible solution mentioned earlier, how can we refine the search process by
    reasoning?

– The second issue can be briefly described as follows. A default setting for searching is
  that if the input query constraint is the same, the results should be exactly the same. The
  diversity of user needs (e.g. the completeness of the query results and the specificity of
  an answer) is not taken into consideration. The following question then arises:

  – Although the same query input is provided, do users with different needs expect the
    same answers?

In order to remove the barriers for large-scale queries, we need to design more concrete
solutions to implement the unification of search and reasoning.

Granular computing, a field of study that aims at extracting the commonality of human
and machine intelligence from the viewpoint of granularity [38,39], emphasizes that human
can always focus on appropriate levels of granularity and perspectives, ignoring irrelevant
information in order to achieve effective problem-solving [39,43]. This process contains
two major steps, namely, the search of relevant data and problem-solving based on searched

data. As a concrete approach for problem-solving based on large-scale data, the unification of search and reasoning also contains the two steps, namely, the search of relevant facts and reasoning based on rules and searched facts. A granule is a set of elements that are drawn together by their equality, similarities, indistinguishability from some aspects (e.g. parameter values) [37]. Granules can be grouped into multiple levels to form a hierarchical granular structure, and the hierarchy can also be built from multiple perspectives [39]. Thus, various hierarchical processing can be performed on the organized structures [6]. Following the above-mentioned inspirations, the scattered large-scale data can be grouped together as granules in different levels or under different views to meet various user needs. From the perspective of granularity, we provide various strategies for unifying user-centric search and reasoning under time constraints to tackle the above issues.

For the first issue stated earlier, the results from the vague or incomplete query form a very huge granule, and they can be further divided into smaller granules in finer levels. All the levels form a hierarchical granular structure. According to the basic-level advantage in cognitive science [30,34], we propose a "starting point" that consists of an input query together with additional constraints derived from a user profile (such as users interests, etc. They help the search process locate to an appropriate level that is convenient to accelerate the process of finding useful results). Furthermore, we emphasize that the search process should begin from the starting point. In this paper, we provide some interest retention models that capture the dynamic shift of user interests based on cognitive memory retention models, then through a reasoning process, we use the acquired interests to construct the starting point and utilize it to refine the query process.

For the second issue stated earlier, we emphasize that the expectations of knowledge query results vary among different users and can be satisfied in different levels of completeness, specificity, and from different perspectives. Several granular computing-inspired strategies are proposed and discussed in detail. From the multi-level point of view, the query can be processed in multiple levels of completeness and multiple levels of specificity. If the user can be satisfied with a lower level of completeness, there is no need to go further to higher levels, since it needs much processing time. If a user's need can be satisfied at a very abstract level (with coarser grain size, namely, granularity), it is not necessary to provide results in more specific levels. From the multi-perspective point of view, a query can be processed based on different viewpoints. If a user's need cannot be satisfied in one perspective, it is rational to change to other perspectives to seek for possible answers.

In order to concentrate on the demonstration of proposed methods, we focus on our discussion in the field of query refinement and processing on large-scale scientific literatures. We carried out some experiments using the SwetoDBLP dataset, an RDF (Resource Description Framework)[1] version of the DBLP Computer Science Bibliography dataset [1]. These results show that the proposed approach can reduce and refine the whole process of querying on large-scale data and provide better results.

This paper is extended from two of our previous papers [40,41]. Section 2 introduces some models that are used to acquire users' retained interests. Section 3 describes granular organizations of knowledge structures as a foundation for query-processing strategies introduced in later sections. Various strategies for user-centric unification of search and reasoning are discussed in the next two sections. Based on user interests retention, Sect. 4 introduces the starting point strategy. Several granular computing-inspired strategies are introduced in Sect. 5, namely, the multi-level completeness strategy in Sect. 5.1, the multi-level specificity strategy in Sect. 5.2, and the multi-perspective strategy in Sect. 5.3. We also provide a

---

[1] http://www.w3.org/RDF/.

preliminary study on the integration of different strategies in Sect. 5.4. Section 6 discusses some related works. Finally, Sect. 7 concludes the paper by highlighting major contributions and describing the future work of this study.

2 User interests retention modeling

User interests can be obtained from either user logs (they may be stored locally or can be tracked through user accounts) or users' public information that they may not mind sharing). In this section, we present some interests retention models to track users' retained interests and examine them on a semantic dataset.

2.1 Interests retention models

User interests can be described as a set of concepts that users are familiar with. Interests retentions may be related to user recent interests and possible queries from users. For simplicity, we first provide a model of measuring interest retention based on the cumulative interest (CI) value:

$$CI(i, n) = \sum_{j=1}^{n} m(i, j) \tag{1}$$

where n is the number of time intervals (e.g. year) considered, $j \in \{1, \ldots, n\}$ is a variable that is used to index each time interval, $m(i, j)$ is the number of appearances of the interest i in the time interval j, and $CI(i, n)$ reflects the value of total interest on i, namely, how many times has an interest appeared in the considered time intervals. The above-mentioned computation may not correctly reflect a researcher's current interests. For example, he/she has shifted the interest, but the accumulated number of an old interest may still be higher than that of a new interest.

Interests retention is to some extent similar to memory retention in cognitive psychology. The loss of interest in an area can be regarded as forgetting of a previously interested topic. We adopt the latter to model the retention of interests. Memory retention models can be categorized into two types, the exponential law-based ones [14,22] and the power law-based ones [28,34].

The forgetting curve proposed by Ebbinghaus [14] is for describing the forgetting mechanism of memory. Loftus suggests that the forgetting function satisfies an exponential formula $P = Ae^{-bT}$, where P represents the performance measure of memory retention, A and b are two parameters for the model, and T is the delay time [2,22]. We assume that the decaying mechanism of interests can be described by a similar formula. In the context of interests, P represents the value of retained interest, and T is the delay time of this interest. Within a time interval, an interest may appear several times, and each appearance of the interesting topic will have an activation for the overall retained interests. Hence, the retained interest value should be a sum of retention values from each previous appearance. Based on the upper formula, the retention of an interest (i) can be denoted as:

$$ERI(i, n) = \sum_{j=1}^{n} m(i, j) \times Ae^{-bT_{i,j}} \tag{2}$$

where $T_{i,j}$ denotes the delay time of the interest i starting from the time interval j. For each time interval j, i might appear $m(i, j)$ times, and each time will contribute a value $Ae^{-bT_{i,j}}$

to the total retention of an interest, where A and b are constants. The value $ERI(i, n)$ is the retention of an interesting topic i through all the time intervals based on the exponential law model.

Recent studies claim that the memory retention can be modeled based on a power function [2,34]. Hence, another model for interests retention can be represented as:

$$PRI(i, n) = \sum_{j=1}^{n} m(i, j) \times AT_{i,j}^{-b} \qquad (3)$$

where $T_{i,j}$ is the delay time of topic i starting from the time interval j. For each time interval j, i might appear $m(i, j)$ times, and $m(i, j) \times AT_{i,j}^{-b}$ is the total retention of an interest contributed by that time interval.

To sum up, $CI(i, n)$ reflects a user's interest on topic i through all the time intervals, which concentrates on frequency of interests and reflects the cumulative interest value through all the considered time. $ERI(i, n)$ and $PRI(i, n)$ concentrate on both frequency and recency. They reflect a user's retained interest on the topic i in a specific time (after some time intervals).

As an illustrative example, we consider a scenario of tracking the authors' research interests, which are implicitly embedded in their own publication lists. In this study, we use the SwetoDBLP dataset [1], an RDF version of the DBLP dataset. The time interval considered is one year (since DBLP does not contain information related to finer time intervals). From the publication lists of authors, it is very hard to acquire the actual values of retained interests, but users' current interests might be related to their retained interests. Hence, we use the values from interests retention models to predict users' current research interests. In this way, we can see whether proposed interest retention models are effective to acquire the retained values for interests.

In our study, in order to minimize the value of ρ in t-test, as a first try, the parameters in the power law model are chosen as $A = 0.855$ and $b = 1.295$. For the calculation of correlation coefficient, Spearman's rank order correlation coefficient was considered [27], because there were a lot of tied items in each dataset (i.e. prediction value dataset and actual value dataset), we actually used the following Pearson's correlation coefficient between ranks [32]:

$$\gamma = \frac{n(\sum x_i y_i) - (\sum x_i)(\sum y_i)}{\sqrt{n\left(\sum x_i^2\right) - \left(\sum x_i\right)^2}\sqrt{n\left(\sum y_i^2\right) - \left(\sum y_i\right)^2}} \qquad (4)$$

where n is the number of items in each dataset, x_i and y_i are the ranks of ith items in each dataset. The same rank (an average of their positions in the ascending order of the items) is assigned to each of the items with an equal value. Hence, the value for the correlation coefficient between the prediction and the real data is $\gamma \approx 0.107$, and for 1-tail t-test, $\rho = 0.237$. For the exponential law model, the parameters are chosen as $A = 0.535$ and $b = 0.382$. The rank correlation coefficient is $\gamma \approx 0.168$, and for 1-tail t-testing, $\rho = 0.129$. The results are, to some extent, close to statistical significance. In order to test the parameters in larger range, in our initial work, we choose the authors whose number of publications are above 100 based on the SwetoDBLP dataset (1,226 authors in all). Using the power functions and relevant parameters introduced earlier, we extract top 9 interests from their interest lists from the year 2000 to 2007 (hence, 1226×8 sets of data are involved). A comparative study on the actual interests and predicted interests has been done. According to the experimental results, 0.98% of the prediction can match at least 7 interests, 3.22% can match 6 interests, 8.35% can match 5 interests, 15.66% can match 4 interests, and 21.33% can match 3 interests. Hence, 49.54% of the predictions can match at least 3 interests in the top 9 interests in our experiment. From

this experimental result, we can conclude that the interests retention has an effect on current interests. To some extent, the values from our interests retention models have some degree of relation with the actual publication numbers (reflecting current research interests), even though there is some gap from statistically significant for the test of Pearson's rank order correlation coefficient between the model prediction results and the real data. The major reasons might be:

– Although interests and publications are related, there is a lag between authors' current interests and what they have published. However, we need to assume that they are very related if we want to model user interests retention based on the publication data.
– A burst of interest in a certain topic has not been considered in the model (e.g. the author might find that many problems need to be solved and he/she could solve them in a relatively short period of time). The traditional model for memory retention also cannot predict exceptions, such as the events that are hard to forget.
– A sudden loss of interest in an area. It may be caused by the fact that an area is getting very mature and does not have many unsolved problems or because other areas have much more attraction to the author and he/she has no time or interest to work on the previous interesting fields.
– The effect from coauthors. One may have many collaborators in a short period of time in an area. Hence he/she, as a coauthor, may have a burst of publication in that area, but these publications may not truly reflect the major interests of him/herself.
– Other environmental factors, such as a sudden event or a breakthrough in a field, may attract some authors working on related topics. No matter they have continuous interests in these areas.

However, the experimental results have shown that at least we have given some applicable models, which can partially reflect the interests retentions and the change of interests. We are still working on this issue for better predictions. Although the accuracy of current models are not ideal, it is enough to prove that interests retentions is one of the major factors, which are related to users' current interests. Hence, it can be utilized to accelerate and refine the search process.

2.2 Case studies of user interests models

A comparative study of Zhisheng Huang's cumulative research interests and current research interests (through the values of interests retention by an exponential law ($A = 0.535$ and $b = 0.382$) model and a power law ($A = 0.855$ and $b = 1.295$) model) are shown in Table 1; Figs. 1 and 2, respectively. We can see that by using the two models, users' recent interests can be obtained.

From Table 1, we can observe that for some research interests, even if they have a big value on the total research interest through Zhisheng Huang's research life (up to now), they may not be his current major research interests. Taking "Agents" as an example, it has the highest value of cumulative research interests but has very low current research interests based on the computation of his research interests retention. Although it is in the third place of the cumulative research interests, "Ontology" is the number 1 current research interest based on both the power law model and the exponential law model. The results shown in Fig. 2; Table 2 are obtained by using the power function-based retained interest model ($PRI(i, n)$). We can observe that there are two characteristics for interests retentions:

– For each publication, its contribution to the overall retained interest value decreases as time passed;

Fig. 1 A comparative study of cumulative interests from 1990 to 2008 and retained interests values in 2009 (based on both the power law model and the exponential law model)
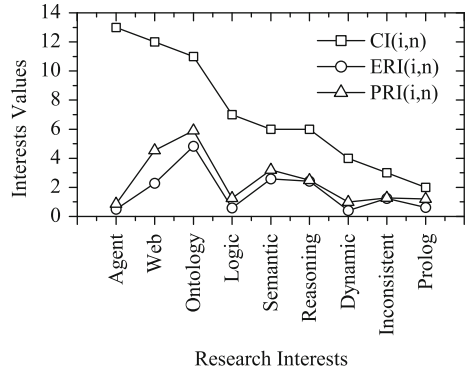
Fig. 2 The contribution values to retained interests (using $PRI(i, n)$) from papers in a certain topic published in different years
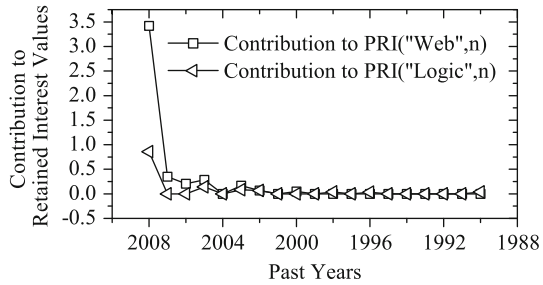
Table 1 A comparative study of cumulative research interests and current research interests (2009) of Zhisheng Huang based on the DBLP publication list

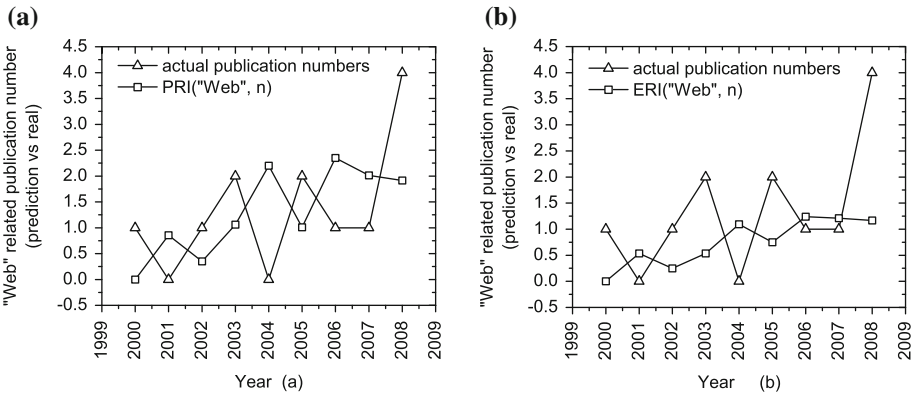| CI(i,n)	PRI(i,n)	ERI(i,n)
Agent(13)	Ontology(5.9041)	Ontology(4.8218)
Web(12)	Web(4.5450)	Semantics(2.5867)
Ontology(11)	Semantics(3.0551)	Reasoning(2.4257)
Logic(7)	Reasoning(2.4845)	Web(2.2742)
Semantic(6)	Prolog(1.2034)	Inconsistent(1.2383)
Reasoning(6)	Inconsistent(1.2672)	Prolog(0.6143)
Dynamic(4)	Logic(1.2567)	Logic(0.5847)
Inconsistent(3)	Dynamic(0.9889)	Agent(0.4921)
Prolog(2)	Agent(0.8741)	Dynamic(0.4112)

– Within the same year, the publication number is positive related with the value of retained interests.

Although not many years later, no matter how many papers an author has published on a specified interest in the past, the effect will be very limited to his/her current research interests. Again, "Agent" is a good example.

Based on the power law model, in order to keep a good result in t-test, we have $b = 1.295$. It shows that recent changes on the interests will have great effects on the interests retention, but the effects will not continue very long. Since previous interests also take some effects, the current interests do not just rely on the most recent appeared terms, as shown in Fig. 3a.

Table 2 Contributions of publications from different years to current research interests based on the power law model ($A = 0.855$, $b = 1.295$)

Year	j	m("Web",j)	Contribution to PRI("Web",n)	m("Logic",j)	Contribution to PRI("Logic",n)
2008	1	4	3.4200	1	0.8550
2007	2	1	0.3484	0	0
2006	3	1	0.2061	0	0
2005	4	2	0.2840	1	0.1420
2003	5	2	0.1680	1	0.0840
2002	6	1	0.0688	1	0.0688
2000	7	1	0.0497	0	0
1998	8	0	0	1	0.0383
1996	9	0	0	1	0.0309
1990	10	0	0	2	0.0378

(a)

(b)

Fig. 3 A comparative study on the prediction and real publication numbers by the power law model (**a**) and the exponential law model (**b**)

In the exponential law model, to keep a good result in t-test, we have $b = 0.382$, and one can observe a similar phenomenon in Fig. 3b.

Although the value of correlation coefficient is not very good, the results from t-test show that to some extent, the retained interests acquired through the proposed models are partially related to users' future interests. Hence, the obtained retained interests can be used to refine the vague or incomplete queries.

3 Granular organization of knowledge structures

A better organization of knowledge is the foundation for effective search and reasoning. In order to search effectively on the large-scale datasets, we need to make good use of the implicit structure of the knowledge sources. We here discuss two types of structures, namely, the hierarchical knowledge structure and the networked knowledge structure.
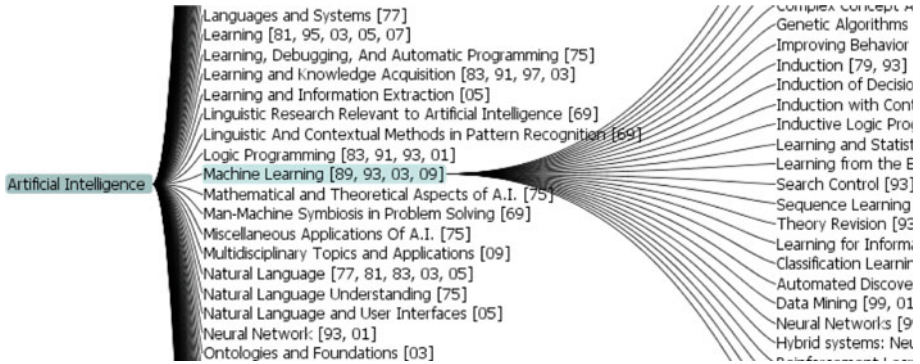
Fig. 4 A partial hierarchical knowledge structure of "Artificial Intelligence" according to an analysis on proceedings indexes of IJCAI 1969–2009
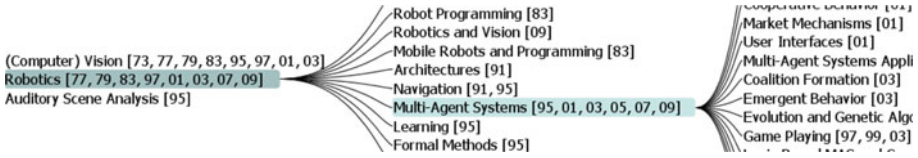
3.1 Hierarchical knowledge structure

Although it does not force a hierarchical organization of knowledge, RDF, as a knowledge representation method for the semantic data implicitly contains the structure in the RDF graph. In this kind of structure, knowledge is represented in multiple grain sizes (i.e. granularities), which provides a multi-level representation of the knowledge sources, with coarser-grained knowledge in a more abstract level and finer-grained knowledge in a more concrete level.

Figure 4 shows a hierarchical knowledge structure of the field "Artificial Intelligence", which is based on an analysis of proceedings indexes of the 1969–2009 International Joint Conferences on Artificial Intelligence (IJCAI). In the context of the semantic data, the hierarchical knowledge structure can be represented using RDF, based on which reasoning task on the hierarchy can be done. The following example shows that "Learning" is a branch field of "Artificial Intelligence", while "Classification" is a branch field of "Learning".

```
<rdf:RDF
    xmlns:rdf=``http://www.w3.org/1999/02/22-rdf-syntax-ns#''
    xmlns:rdfs=``http://www.w3.org/2000/01/rdf-schema#''>
  <rdfs:Class rdf:ID=``Learning''>
    <rdfs:subClassOf rdf:resource=``Artificial Intelligence''/>
  </rdfs:Class>
  <rdfs:Class rdf:ID=``Classification''>
    <rdfs:subClassOf rdf:resource=``Learning''/>
    <rdfs:subClassOf rdf:resource=``Artificial Intelligence''/>
  </rdfs:Class>
</rdf:RDF>
```

We can infer that if one needs very general information with respect to the field "Artificial Intelligence", he/she may just want the knowledge in the second level, which includes around 100 branches of AI (in fact, if we do not organize the index of these proceedings into a hierarchical knowledge structure, one may get around 400 branches, which are very confusing in one level). Furthermore, if he/she needs more detailed knowledge concerning one branch of AI, say "Robotics", he/she can choose "Robotics" and get a finer-grained structure, as shown in Fig. 5. In this way, the knowledge source is provided in different levels of details with an interactive manner concerning different user needs. Later, we will use the characteristics to supervise the unification of search and reasoning.

Fig. 5 Finer-grained subknowledge structure on "robotics" in the hierarchical knowledge structure of "Artificial Intelligence"

(a)   **(b)**
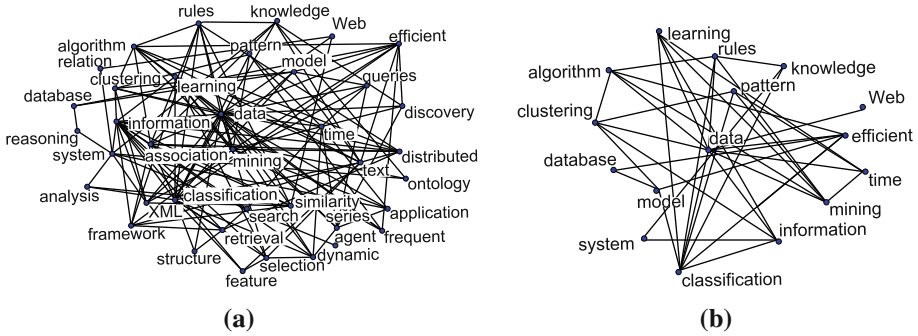
Fig. 6 Simplified networked knowledge structures according to words co-occurrence analysis on literature titles in "Knowledge and Information Systems" (from vol. 1(1) 1999 to vol. 20(2) 2009. In **a** every word appears at least 10 times, while in **b** every word appears at least 20 times)

3.2 Networked knowledge structure

In most cases, especially under the context of the semantic dataset represented using RDF or OWL (Web Ontology Language),[2] the hierarchical knowledge structure is embedded in a networked knowledge structure [3]. It is composed of many concepts that interconnect with each other through various relations. Nodes in the networked knowledge structure can be grouped as different granules according to their values of node degree. If the knowledge structure is too complex for users, instead of showing all the structures, one may choose to show a substructure that only contains pivotal nodes (which have relatively higher node degrees) and their interrelationships. In this way, a simplified but important substructure is provided.

Figure 6a illustrates a simplified co-occurrence analysis of words in the literature titles of the journal, "Knowledge and Information Systems (KAIS)", from vol. 1(1) 1999 to vol. 20(2) 2009. Words are connected with each other by their co-occurrence in the same paper title, which can be considered as a networked knowledge structure (for simplification, here we only consider single word terms). Although it only shows the words that appear at least 10 times in the whole list, Fig. 6a is still very complex. Fig. 6b shows more important nodes in this network (words which appear at least 20 times are selected). We can see that the structure shown in Fig. 6b is much simpler, but the major structure of the more complex one has been illustrated (and it is not hard to find the major hot topics in this journal). In this way, a networked knowledge structure can be represented in multiple levels of importance based on the node degree, and this may provide inspirations for developing methods on searching a better subset of the original dataset for reasoning.

---

2 http://www.w3.org/TR/owl-features/.

For the granular organization of knowledge structures and concerning scalability, in the hierarchical knowledge structure, we provide a way of presenting knowledge in different levels of details, and in the networked knowledge structure, we provide a way of presenting knowledge structures concerning their different levels of importance in the whole structure. However, we have not discussed how these structures can be used. In the next two sections, we will discuss how to use the two types of structures combining the results from Sect. 2 to reduce and refine the search and reasoning process.

4 The starting point strategy

4.1 What is the starting point?

In query tasks, as for the question "where to start a query?", we observe that a query process may not just start with the user inputs in the query box, because sometimes the query is not precise. With the original query, the system may provide a huge amount of irrelevant information. For example, when a machine learning researcher searches "Tom" on Google, he/she may get more than 496000,000 webpages containing "Tom Hanks", "Tom Cruise", "Tom and Jerry", and "Tom Mitchell", etc. What he/she really needs may be just those webpages that contain "Tom Mitchell" (around 31,600 webpages). The user has to refine the query or try to find what he/she really wants in a list of query results with many irrelevant ones.

The results from the vague or incomplete query ("Tom") form a relatively large granule in a coarser level, and they can be further divided into smaller granules in finer levels (such as a set of search results for "Tom Hanks", "Tom Mitchell", etc. separately). Thus, a hierarchical granular structure is formed based on these levels of results. Psychological experiments have proofed that human prefers to solve problems using terms in the basic level (the ones that are used more frequently than others [35]), and in this way the problem-solving process can be accelerated [30]. In the upper example, "Tom Mitchell" can be considered as a term in the basic level for machine learning researchers. Following the results from cognitive psychology, we assume that, in our study, user interests are closely related to a user's basic level. We propose to utilize the basic level in the result hierarchy to find the most relevant search results.

Based on the hierarchical granular structure and inspired by the basic level for human problem-solving, we define a *starting point, SP* that consists of a user identity (e.g. a user name, a URI, etc.) and a set of nodes that serve as the background for the user (e.g. user interests, friends of the user, or other user familiar or related information). Hence, the retained interests that are obtained through interest retention models proposed in Sect. 2.1 can be considered as some nodes in the SP.

4.2 Constructing a starting point

As mentioned in Sect. 4.1, in this paper, we consider a starting point is composed of a user ID and several nodes, which reflect the user interests. User interests can be obtained based on the interest retention models proposed in Sect. 2.1, and together with the user ID, they can be organized as an RDF file that serve as additional data to the original dataset.

To illustrate our idea, we continue our experiment using the SwetoDBLP dataset. Since the user interests are not described in the original dataset, we developed a user interests extraction system (a subsystem of the DBLP search support engine (DBLP-SSE)[3]), which

---

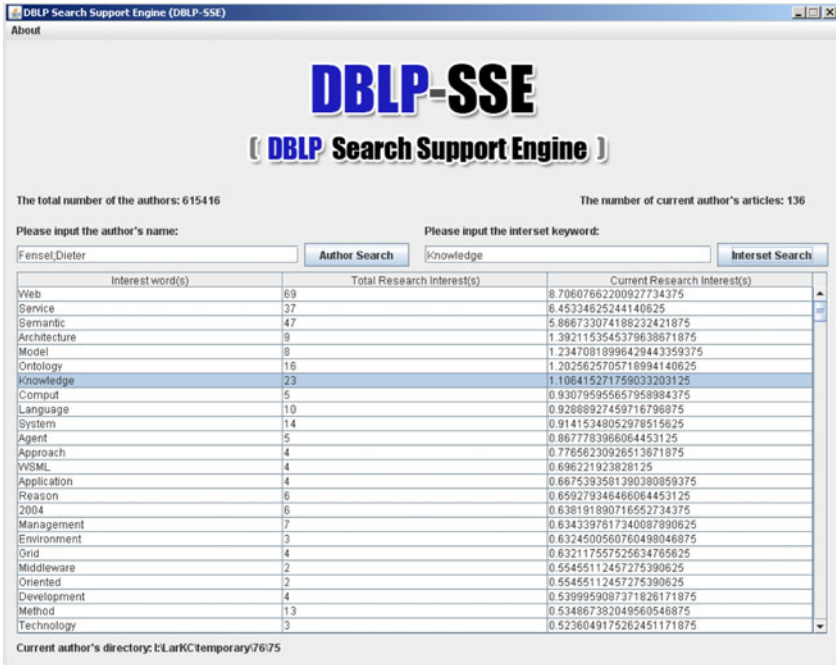[3] http://www.wici-lab.org/wici/dblp-sse.

Fig. 7 Computer scientists retained interests extraction from SwetoDBLP

extracts the information of all the authors (including author names, publication lists, and corresponding years) from the SwetoDBLP dataset and reports their retained interests based on the power law model and the exponential law model introduced in Sect. 2.1. Given the corresponding name, one can search any authors' retained interests based on their previous publications. One can also search on a specified interest and get the corresponding value of that interest. A screen shot of the system is shown in Fig. 7.

The observed interests for each author are represented as an RDF file and added to the original dataset, so that they can be used in the future query process. Vocabularies from the FOAF (Friend of a Friend) project[4] are used in our study. We keep 9 interest terms for each author [24], since in most cases, it will be very hard for an author to hold a large number of interests at the same period of time. Furthermore, the picking of interests is based on their ranking from the calculation of interests retention.

The following is a partial sample RDF file representing Zhisheng Huang's current research interests (using the power law model) through an analysis of his publications from 1990 to 2008:

```
<rdf:RDF
 xmlns:rdf=''http://www.w3.org/1999/02/22-rdf-syntax-ns#''
 xmlns:foaf=''http://xmlns.com/foaf/0.1/''>
 <foaf:Person rdf:about=''http://www.informatik.uni-trier.de/
 ~ley/db/indices/a-tree/h/Huang:Zhisheng.html''>
   <foaf:name>Zhisheng Huang</foaf:name>
   <foaf:topic_interest>
     <rdf:Seq>
```

---

[4] http://www.foaf-project.org/.

```
         <rdf:li>Ontology</rdf:li>
         <rdf:li>Web</rdf:li>
         <rdf:li>Semantics</rdf:li>
         <rdf:li>Reasoning</rdf:li>
      </rdf:Seq>
   </foaf:topic_interest>
 </foaf:Person>
</rdf:RDF>
```

where we use foaf:topic_interest to describe the author's interests and the RDF sequence container to describe the order of interests. If Zhisheng Huang posts a query, the constraints from his interests can be added to the original query as additional constraints for refinement. We have released the computer scientists research interest RDF[5], which includes current research interests of all computer scientists listed in the SwetoDBLP dataset, so that other researchers could create their own application using this semantic dataset.

4.3 Refining search by reasoning with a starting point

Based on the idea of starting point, the search of important nodes for reasoning can be based on the following principles:

– Principle 1 (Familiarity driven): The search process first selects out the nodes that are directly related to the SP, which is relevant to the query, and SP related results are ranked to the front of others.
– Principle 2 (Novelty driven): The search process first selects out the nodes that are not directly related to a SP, which is relevant to the query, and the SP-related results are pushed to the end of others.

Principle 1 is designed to meet the needs from the users who want to get more familiar results first. Principle 2 is designed to meet the needs from those who want to get unfamiliar results first. One example for Principle 2 is that in news search, the users, in most cases, want to find the relevant news information that has not been visited. Using these two principles, several subsets of the original dataset can be selected out for reasoning. Hence, based on the starting point, a strategy that unify search and reasoning together is proposed.

The starting point strategy is outlined as the following major steps:

Step 1. Choose top N interests that have the biggest values of interest retentions based on one of the proposed models ($CI(i, n)$, $ERI(i, n)$ or $PRI(i, n)$), and construct a SP based on selected interests.
Step 2. Judge whether the query input by the user contains all the information in the SP. If yes, go to *Step* 4.
Step 3. Rewrite the vague or incomplete query using the SP.
Step 4. Query the dataset using the (rewritten) query based on the familiarity-driven principle or the novelty-driven principle.

If the query system can acquire user-related information (e.g. name, URI, etc.) and some information from which user interests may be extracted (e.g. a publication list, visited pages, etc.), then it starts these steps. We should emphasize that not all the queries need to be refined, only those which do not contain constraints from user background should be considered for refinement.

---

[5] http://www.wici-lab.org/wici/dblp-sse.

Fig. 8 A DBLP search support engine (DBLP-SSE)

Since in most cases, queries are very related to user interests contained in the starting point, the vague query can be refined with the user interests to help users get a more relevant set of query results. In the field of information retrieval, this process can be considered as one type of query extension. However, there are some reasoning tasks during this process, which has been implicitly done by the search program developers. Nevertheless, it is worthy of emphasizing this process as refining search by reasoning [7]. The reasoning process can be described using the following rule:

$$hasInterests(U, I), hasQuery(U, Q), executesOver(Q, D), \neg contains(Q, I) \rightarrow$$
$$refinedAs(Q, Q'), contains(Q', I), executesOver(Q', D).$$

where $hasInterests(U, I)$ represents that the user "U" has a list of interests "I" and can be acquired. $hasQuery(U, Q)$ represents that there is a query input "Q" by the user "U". $executesOver(Q, D)$ denotes that the query "Q" is executed over the dataset "D". $\neg contains(Q, I)$ represents that the query "Q" does not contain the list of interests "I". $refinedAs(Q, Q')$ represents that the original query "Q" is refined by using the list of interests as "Q'". $contains(Q', I)$ denotes that "Q'" contains the list of interests "I". $executesOver(Q', D)$ represents that the refined query "Q'" executes over the dataset "D".

Based on the introduced method, we developed a DBLP search support engine (DBLP-SSE).[6] This system allows an author to log in using his/her own name, which is consistent in the DBLP publication list, then the system will generate a series of interest keywords from his/her own publication list according to the introduced interests retention models. When the user inputs a query in the search box, the system will automatically add constraints using the
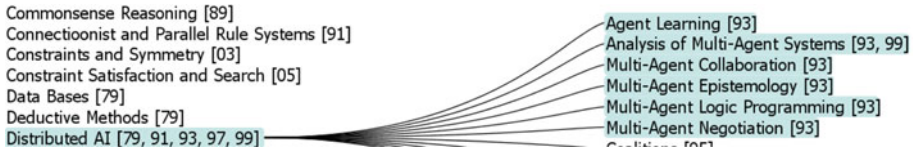
---

[6] http://www.wici-lab.org/wici/dblp-sse/.

Table 3 A comparative study of search results without and with a starting point

Log in name	Dieter Fensel
Top 9 interests	Web, service, semantic, architecture, model, ontology, knowledge, computing, language
Query :	Artificial intelligence
List 1 :	without a starting point (which contains current interests)
	* PROLOG Programming for *Artificial Intelligence*, Second Edition.
	* *Artificial Intelligence* Architectures for Composition and Performance Environment.
	* *Artificial Intelligence* in Music Education: A Critical Review.
	* Music, *Intelligence* and Artificiality. *Artificial Intelligence* and Music Education.
	* Musical Knowledge: What can *Artificial Intelligence* Bring to the Musician?
	* Readings in Music and *Artificial Intelligence*.
	* *Artificial Intelligence* Techniques in Medicine and Healthcare.
	* *Artificial Intelligence* in the HyperClass: Design Issues.
	* Essentials of *Artificial Intelligence*.
	* ……
List 2 :	with a starting point (which contains current interests)
	* **Web** *Intelligence* and *Artificial Intelligence* in Education.
	* *Artificial Intelligence* Exchange and **Service** Tie to All Test Environments (AI-ESTATE)-A New Standard for System Diagnostics.
	* **Semantic Model** for *Artificial Intelligence* Based on Molecular Computing.
	* Open Information Systems **Semantics** for Distributed *Artificial Intelligence*.
	* *Artificial Intelligence* and Financial **Services**.
	* *Artificial Intelligence* Techniques in Retrieval of Visual Data **Semantic** Information.
	* **Semantic** Optimization in Data Bases Using *Artificial Intelligence* Techniques.
	* *Artificial Intelligence* for **Semantic** Understanding.
	* Natural **Semantics** in *Artificial Intelligence*.
	……

top 9 interests from the user interest list to the original query. For each query input by a user, the system generates two lists of search results. The first list is produced using the original query, and the second list is produced based on the refined query. A screen shot of the system is shown in Fig. 8. Furthermore, Table 3 gives an example produced by this system. After logging in using the name "Dieter Fensel", the user provides a query input, and different results are presented according to the original query and the refined query. List 1 is a partial list without the retained interests constraints (namely, without a starting point), while List 2 is the one with them. As we can see that if he wants to find useful results from List 1, a long period of time may be needed for browsing to find more relevant ones in the list. While the one with interest constraints, List 2, is much closer to the user's interests, and the interest constraints do not need to be added manually into the original query. We can observe that as indicated by Berners-Lee [7], the reasoning process helps to refine the query and get a more relevant and correct set of search results.

This example shows how a smaller query results subset in a finer level is acquired without the necessity of finding all the results in a coarser level, which are obtained based on the vague query. As shown in Table 3, the input query term is "Artificial Intelligence" and if we only use this term, the query is too general and we will get too many query results. Those

Commonsense Reasoning [89]
Connectioonist and Parallel Rule Systems [91]
Constraints and Symmetry [03]
Constraint Satisfaction and Search [05]
Data Bases [79]
Deductive Methods [79]
Distributed AI [79, 91, 93, 97, 99]

Agent Learning [93]
Analysis of Multi-Agent Systems [93, 99]
Multi-Agent Collaboration [93]
Multi-Agent Epistemology [93]
Multi-Agent Logic Programming [93]
Multi-Agent Negotiation [93]

Fig. 9 Automatic navigation to the "basic level" that may contain more interesting contents ("Agents" which has the highest value in TI) for Zhisheng Huang

which meet user needs may not appear in the top ones (as shown in List 1). User interests act as constraints that help to find a more relevant subset of query results. In this example, when combining "Artificial Intelligence" with the user interests, such as "Web, Service, Semantics, Knowledge, etc.", it actually goes into some finer levels (such as "Artificial Intelligence + Web") of the result hierarchy, and in this way, the results that are relevant to user interests are ranked in the top ones for user inspection.

We here provide another example that uses the starting point strategy on a hierarchical knowledge structure. Taking Zhisheng Huang as an example again, if he wants to use the multi-level knowledge structure of "Artificial Intelligence" based on our analysis on proceedings indexes of the 1969–2009 IJCAI conferences, what he wants to know may be not branches of AI in general (which are shown in the second level), but something related to his research interests. Based on his cumulative interests ($CI(i, n)$) developed in Sect. 2, the system may suggest a substructure as shown in Fig. 9. The structure has been navigated to the third level that contains much more information on "Agent" that Zhisheng Huang may have much interest according to the highest value in $CI(i, n)$.

For the same query, the starting point strategy realizes the diversity of user backgrounds. The query process is refined through the reasoning process of adding additional constraints from user interests. In the next section, we will try to satisfy different user needs using some strategies inspired by some basic thoughts from granular computing. The multi-level completeness strategy, the multi-level specificity strategy, and the multi-perspective strategy are going to be discussed.

5 Granular computing-inspired strategies

As introduced in Sect. 1, the fundamental idea of granular computing offers a concrete approach that unifies search and reasoning together. In the study of granularity in human and machine intelligence, the core idea is multiplicity, namely, multi-level and multi-perspective [4,17,25,39,43]. In this section, we present several such strategies to solve the proposed problems based on the granular organization of the data and combining with the starting point strategy as stated in Sects. 3 and 4, respectively.

5.1 The multi-level completeness strategy

Large-scale reasoning is very hard to achieve complete results, since the user may not have time to wait for a reasoning system going through the complete dataset. If the user does not have enough time, a conclusion is made through reasoning based on a searched partial dataset, and the completeness is not very high since there are still some sets of data that remain to be unexplored. If more time is allowed and the reasoning system can get more subdatasets

through search, the completeness can migrate to a new level since the datasets cover wider range. There are two major issues in this kind of unifying process of search and reasoning:

1. Since under time constraint, a reasoning system may just can handle a subdataset, methods on how to search for an appropriate subset need to be developed.
2. Since this unification process requires that user judges whether the completeness of reasoning results is good enough for their specific needs, a prediction method for completeness is required.

We name this kind of strategy as unifying search and reasoning with *multi-level completeness*, which provides reasoning results in multiple levels of completeness based on the searched subdataset under time constraints. Moreover, it also provides prediction on the completeness value for user judges.

For the first issue, searching for a more important subdataset for reasoning may be a practical approach to select the subset effectively [15]. The RDF dataset is composed of triples, which contain a set of nodes (subjects and objects) and a set of relations (predicates) on these nodes (namely, the RDF datasets have networked knowledge structures). Hence, in this study, we borrow the idea of "pivotal node" from network science [5], and propose a network statistics–based data selection method. Under this method, we use the node degree (denoted as $degree(n)$, where n is a node in the RDF graph) to evaluate the importance of a node in a dataset (as briefly mentioned in Sect. 3.2). The nodes with relatively high value of node degrees are selected as more important nodes and grouped together as a granule for reasoning tasks. There might be many types of predicates that are associated with the nodes in the RDF dataset, and different meanings of various predicates may meet user needs from different perspectives. According to a specific need from a perspective, (which will be explained in detail in Sect. 5.3), we choose one type of predicate to investigate on the importance of a node. When we only consider this type of predicate and neglect other types, a subgraph of the original RDF dataset can be selected out. In this subgraph, the node degree considering a special type of predicate p can be denoted as $degree(n, p)$.

For the second issue, we here give a formula to produce the predicted completeness value ($PC(i')$) when the nodes that satisfy $degree(n, p) \geq i'$ (i' is a nonnegative integer) have been involved.

$$PC(i') = \frac{|N_{rel(i')}| \times (|N_{sub(i')}| - |N_{sub(i'')}|)}{|N_{rel(i')}| \times (|N| - |N_{sub(i'')}|) + |N_{rel(i'')}| \times (|N_{sub(i')}| - |N|)} \qquad (5)$$

where $|N_{sub(i')}|$ represents the number of nodes that satisfy $degree(n, p) \geq i'$, $|N_{rel(i')}|$ is the number of nodes that are relevant to the reasoning task among the involved set of nodes $N_{sub(i')}$, and $|N|$ is the total number of nodes in the dataset. Using Fig. 10, we give an explanation on how we formulate the prediction function.

The basic idea is that, before the end of the processing task, when all the nodes that satisfy $degree(n, p) \geq i'$ have been processed, we estimate how many results are there over all, and the ratio of completeness is provided based on the number of acquired results (when stopping at $degree(n, p) = i'$) and the number of predicted results. We first can obtain a linear function which goes through $(|N_{sub(i')}|, |N_{rel(i')}|)$ and $(|N_{sub(i'')}|, |N_{rel(i'')}|)$ (i'' is the last assigned value of $degree(n, p)$ by the user for stopping the reasoning process before i'), as shown in Fig. 10. The linear function has a intersection point with the strait line $|N_{sub(i)}| = |N_{sub(0)}| = |N|$ ($|N|$ is the number of nodes in the whole dataset and only needs to be acquired once. It can be calculated offline), and the value for $|Np\_rel(0)|$ is the predicted number of results in the whole dataset (notice that this number always satisfies $|Np\_rel(0)| \leqslant |N|$). Then the predicted completeness value can be acquired.
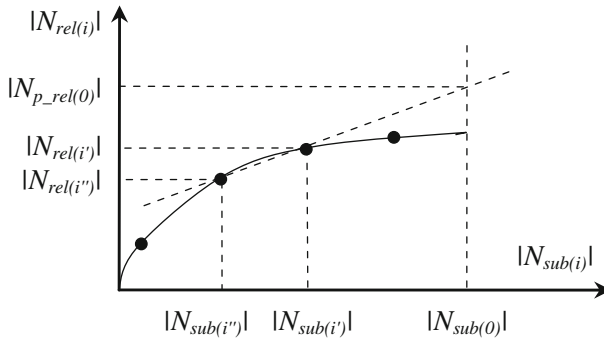
Fig. 10 The schematic diagram for completeness prediction

The multi-level completeness strategy can be described as the following major steps:

Step 1. Calculate $degree(n, p)$ for each node and the number of nodes $|N|$.

Step 2. Show the degree range to the user and ask for the degree to stop.

Step 3. Start the query process from the nodes with the biggest degree, gradually reduce the node degree.

Step 4. Stop at the node degree that the user specifies (if the user does not specify the degree to stop, then stop at $degree(n, p) = 0$).

Step 5. Predict the completeness and ask whether the user is satisfied with the results and the predicted completeness. If satisfied (or $degree(n, p) = 0$), then stop the query process. Else, ask for a smaller degree to stop and go to *Step 6*.

Step 6. Continue the query process from the degree where has been stopped, gradually reduce the node degree and go back to *Step 4*.

As an illustrative example, we take the task "Who are authors in Artificial Intelligence (AI)?" based on the SwetoDBLP dataset. For the simplest case, the following rule can be applied to find relevant authors:

$$has Paper(U, P), has Title(P, T), contains(T, \text{"Artificial Intelligence"}) \rightarrow$$
$$author(U, \text{"AI"}).$$

where $haspaper(U, P)$ denotes that the author U has a paper P. $has Title(P, T)$ denotes that P is titled T. $contains(T, \text{"Artificial Intelligence"})$ denotes that the title T contains the term "Artificial Intelligence". $author(U, \text{"AI"})$ denotes that the author U is an author in the field of Artificial Intelligence.

Since the SwetoDBLP dataset contains too many publications (more than 1200,000), doing reasoning based on a dataset like this may require an unacceptable period of time, it is better that more important authors could be provided to the user first. We here choose the predicate that indicates an author has a coauthor (denoted as p_{cn}). Under this perspective, the authors with more coauthors, namely, having a higher value of $degree(n, p_{cn})$, are more important. In order to illustrate the levels of completeness, we randomly choose some $degree(n, p_{cn})$ to stop the reasoning process, as shown in Table 4. The reasoning process will start from the nodes with the biggest value of $degree(n, p_{cn})$, reduce the value gradually as time passed by and will stop at the chosen $degree(n, p_{cn})$ for user judges. In order to meet users' specific needs on the levels of completeness value, one can use the proposed completeness prediction method as introduced earlier, and the prediction value has also been provided in Fig. 11. This prediction value serves as a reference for users to judge whether they are satisfied. If more

Table 4 Unifying search and reasoning with multi-level completeness and anytime behavior

$degree(n, p_{cn})$ value to stop	Satisfied authors	AI authors
70	2,885	151
30	17,121	579
11	78,868	1,142
4	277,417	1,704
1	575,447	2,225
0	615,124	2,355

Fig. 11 Comparison of predicted and actual completeness value
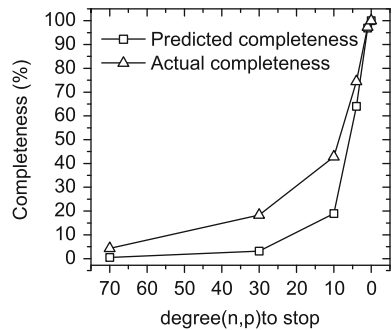
time is allowed and the user has not been satisfied yet, more nodes are involved, and the user is supposed to get reasoning results with higher levels of completeness.

5.2 The multi-level specificity strategy

Reasoning results can be either very general or very specific. If the user has not enough time, the search and reasoning process will just be on a very general level. And if more time is available, this process may go to a more specific level, which contains results in a finer level of grain size (granularity). Namely, the unification of search and reasoning can be with *multi-level specificity*, which provides reasoning results in multiple levels of specificities under time constraints.

The study of the semantic networks emphasizes that knowledge is stored as a system of propositions organized hierarchically in memory [12]. The concepts in various levels are with different levels of specificities. Hence, the hierarchical knowledge structure can be used to supervise the unification of search and reasoning with multi-level specificity. In this process, the search of subdatasets is based on the hierarchical relations (e.g. sub-class of, subproperty of, instance of, etc.) among the nodes (subjects and objects in RDF) and is forced to be related with the time allowed. Nodes that are not subclasses, instances or subproperties of other nodes (namely, semantically the most general nodes) will be searched out as the first level for reasoning. If more time is available, deeper levels of specificity can be acquired according to the transitive property of these hierarchical relations. The specificity will just go deeper for one level each time before the next checking of available time (nodes are selected based on direct hierarchical relations with the nodes from the former direct neighborhood level).

The multi-level specificity strategy can be summarized as the following major steps:

Step 1. Judge whether the query results are with multi-level specificities through finding out the hierarchical knowledge structure (within the dataset or from other sources) that is related with the query.

Step 2. Start the query from the data with the coarsest level of specificity.

Step 3. Judge whether there is more processing time. If not, then stop the query.

Step 4. Ask whether the user satisfies the results. If yes, then stop the query.

Step 5. Query on the level of data that is one level finer than the previous level (if it is the finest level, then stop after query on this level) and go back to *Step* 3.

As an illustrative example, we use the same reasoning task in the previous section. For the very general level, the reasoning system will just provide authors whose paper titles contain "Artificial Intelligence", and the reasoning result is 2,355 persons (it seems not too many, which is not reasonable. Since in many cases, the authors in the field of AI do not write papers whose titles include the exact term "Artificial Intelligence", they may mention more specific terms such as "Agent", "Machine Learning", etc.). If more time is given, answers with a finer level of specificity according to a hierarchical domain ontology of "Artificial Intelligence" can be provided. Based on all section and subsection names of AI-related conferences in the DBLP, we create a "three-level Artificial Intelligence ontology" automatically (this ontology has a hierarchical structure representing "Artificial Intelligence-" related topics. Topic relations among levels are represented with "rdfs:subClassOf"), and we utilize this ontology to supervise the unification of search and reasoning with multi-level specificity.[7] The rule for this reasoning task can be described as:

$$hasResttime, hasPaper(U, P), hasTitle(P, T), contains(T, H), topics(H, \text{``AI''})$$
$$\rightarrow author(U, \text{``AI''}).$$

where *hasResttime* is a dynamic predicate that denotes whether there is some rest time for the reasoning task[8], *topics*(*H*,"AI") denotes that *H* is a related sub-topic from the hierarchical ontology of AI. If the user allows more time, based on the "rdfs:subClassOf" relation, the subtopics of AI in Level 2 of the ontology will be used as *H* for reasoning to find more authors in the field of AI. Further, if the user wants to get results finer than Level 2, then the subtopics in Level 3 are used as *H* to produce an even more complete result list. As shown in Tables 5 and 6, based on the hierarchy of Artificial Intelligence, in which Levels 2 and 3 contain more specific subbranches, it is not surprising that one can get more authors when deeper levels of terms are considered. Hence, the completeness of the reasoning result also goes to higher levels, as shown in Table 6.

5.3 The multi-perspective strategy

User needs may differ from each other when they expect answers from different perspectives. In order to avoid the failure of understanding in one way, knowledge needs to be represented in different points of view [25]. If the knowledge source is investigated in different perspectives, it is natural that the search and reasoning results might be organized differently. Each perspective satisfies user needs in a unique way. As another key strategy, unifying search and reasoning from *multi-perspective* aims at satisfying user needs in multiple views.

---

[7] Here we ignore the soundness of this ontology, which is not the focus of this paper (supporting materials on how we build the ontology can be found from: http://www.wici-lab.org/wici/user-g.). One can choose other similar ontologies instead.

[8] For implementation, logic programming languages such as Prolog does not allow a dynamic predicate like *hasResttime*. But we can consider *resttime*(*R*) as a counter that would return a number. Then, we can check the number to know whether there is any rest time left. Namely: $resttime(R), R > 0 \rightarrow hasResttime$.

Table 5 Answers to "Who are the authors in Artificial Intelligence?" in multiple levels of specificity according to the hierarchical knowledge structure of Artificial Intelligence

Specificity	Relevant keywords	Number of authors
Level 1	Artificial Intelligence	2,355
Level 2	Agents	9,157
	Automated reasoning	222
	Cognition	19,775
	Constraints	8,744
	Games	3,817
	Knowledge representation	1,537
	Natural language	2,939
	Robot	16,425
	…	…
Level 3	Analogy	374
	Case-based reasoning	1,133
	Cognitive modeling	76
	Decision trees	1,112
	Proof planning	45
	Search	32,079
	Translation	4,414
	Web intelligence	122
	…	…

Table 6 A comparative study on the answers in different levels of specificity

Specificity	Number of authors	Completeness (%)
Level 1	2,355	0.85
Level 1, 2	207,468	75.11
Level 1, 2, 3	276,205	100

For a large-scale data source represented in RDF, the perspectives can be characterized using predicates in the RDF triple. Based on each perspective, a subgraph of the original RDF graph can be selected out, and each subgraph reflects one characteristic of the whole. For simplification, we consider the situation that a perspective is characterized using a single type of predicate. Hence, except for choosing perspectives according to user preference, they can be chosen according to the importance of the predicates, which can be evaluated through their appearance times. Those that appear more frequently can be chosen as more important perspectives.

The multi-perspective strategy can be summarized as the following major steps:

Step 1. Judge whether the user has specified the perspective for query. If yes, then goto *Step* 3.
Step 2. Choose a perspective by appearance times of the predicates.
Step 3. Query processing from the chosen perspective.
Step 4. If the user satisfies with the query results, then stop the query process. Otherwise, ignore the current perspective and go back to *Step* 2 (if time is allowed).
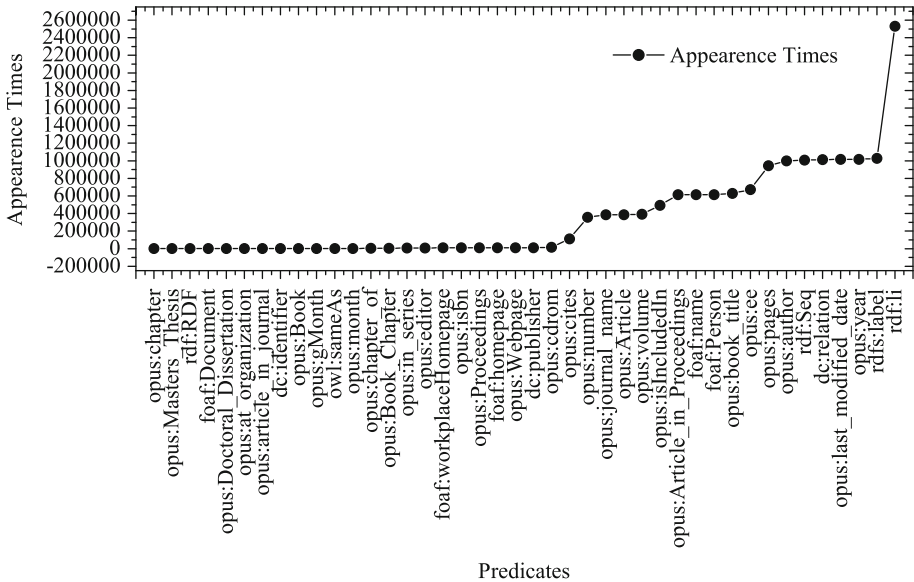
**Fig. 12** Predicates appearance times in the SwetoDBLP RDF dataset

As an illustrative example, we continue the task of "Who are authors in Artificial Intelligence ?". As described earlier, we use node degree under a perspective ($degree(n, p)$) to search for a subset of the original data for reasoning. Figure 12 shows the distribution of appearance times of all kinds of predicates in the SwetoDBLP dataset. According to this figure, we find that among the predicates who hold relatively more appearance times, "rdf:Seq" and "rdfs:label" are very meaningful ("rdf:Seq" can be used to find coauthor numbers, and "rdfs:label" can be used to find publication numbers for each author). Hence, we analyze the distribution of the node degrees under the perspective of coauthor numbers (p_{cn}) and publication numbers (p_{pn}).

First, we choose the perspective of the coauthor numbers. From this perspective, we find the coauthor number distribution characteristics on the SwetoDBLP dataset, as shown in Fig. 13. Figure 13b is the log–log diagram of coauthor number distribution (as shown in Fig. 13a). In the left side of Fig. 13b, there is a peak value, and it does not appear at the starting point or the ending point of the distribution curve. As a comparison of the coauthor number view, we provide the distribution analysis from the view point of publication number (as shown in Fig. 14). We observe that, for the log–log diagram, different from the perspective of coauthor number distribution, the publication number distribution is without a peak value in the middle of the distribution curve, as shown in Fig. 14b.

Table 7 provides a partial result for the experiment using the multi-level specificity strategy introduced in Sect. 5.2 from two perspectives (namely, publication number and coauthor number). As shown in Figs. 13b and 14b; Table 7, it is clear that since the distribution of node degree under the above two perspectives are different, and for the same node, the node degree under these two perspectives are different. Hence, we can conclude that using different perspectives, both of the sequence of nodes provided for reasoning and the reasoning results are organized differently (as shown in Table 7). In this way, various user needs can be satisfied.

(a)

(b)
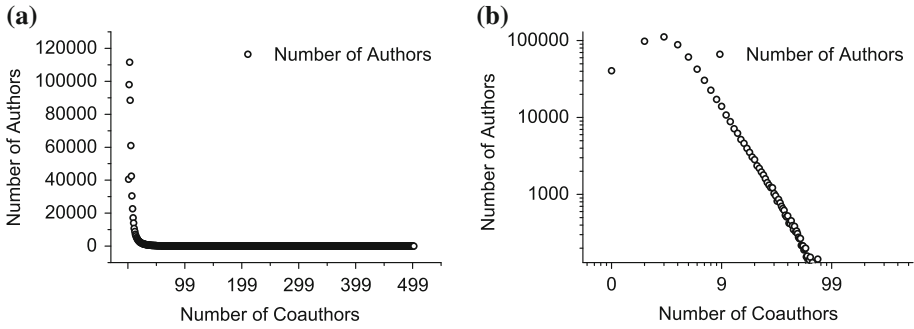
Fig. 13 The coauthor number distribution in the SwetoDBLP dataset **a** and its log-log diagram **b**

(a)

(b)
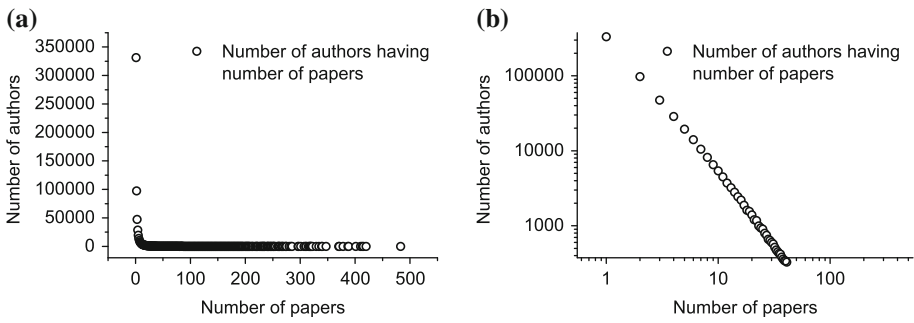
Fig. 14 The publication number distribution in the SwetoDBLP dataset (**a**) and its log-log diagram (**b**)

Table 7 A partial result for the query task "The list of authors in Artificial Intelligence" using the multi-level specificity strategy from two perspectives

Specificity level	Publication number perspective	Coauthor number perspective
Level 1 (Artificial Intelligence)	Thomas S. Huang (387)	Carl Kesselman (312)
	John Mylopoulos (261)	Thomas S. Huang (271)
	Hsinchun Chen (260)	Edward A. Fox (269)
	Henri Prade (252)	Lei Wang (250)
	Didier Dubois (241)	John Mylopoulos (245)
	Thomas Eiter (219)	Ewa Deelman (237)
	…	…
Level 2 (Knowledge Representation)	Elisa Bertino (420)	Elisa Bertino (305)
	Jiawei Han (317)	John Mylopoulos (245)
	Christos H. Papadimitriou (296)	Steffen Staab (236)
	Witold Pedrycz (275)	Hsinchun Chen (234)
	John Mylopoulos (261)	Matthias Jarke (227)
	Hsinchun Chen (260)	Li Li (226)
	…	…

Table 8 A partial result list of "Artificial Intelligence" authors using the multi-level specificity strategy from the merging of two perspectives

Specificity level	Ranking	Author names	r_{pn}	r_{cn}	$R(i)$
Level 1 (Artificial Intelligence)	1	Thomas S. Huang	1	2	1.5
	2	John Mylopoulos	2	5	3.5
	3	Hsinchun Chen	3	8	5.5
	4	Edward A. Fox	9	3	6
	5	Yan Zhang	15	11	13
	6	Gio Wiederhold	16	12	14
	7	Steffen Staab	22	7	14.5
	…	…	…	…	…
Level 2 (Knowledge Representation)	1	Elisa Bertino	1	1	1
	2	John Mylopoulos	2	5	3.5
	3	Jiawei Han	7	2	4.5
	4	Hsinchun Chen	4	6	5
	5	Witold Pedrycz	11	4	7.5
	6	Matthias Jarke	5	11	8
	7	Tharam S. Dillon	9	7	8
	…	…	…	…	…

As a step forward, one can try to merge the results from various perspectives together, so that the effects from multiple perspectives can be considered at the same time. If the results from different perspectives are with rankings, the ranking of the merged results is based on the following function:

$$R(i) = w_{p1}r_{p1}(i) + w_{p2}r_{p2}(i) + \cdots + w_{pn}r_{pn}(i), \qquad (6)$$

$$w_{p1} + w_{p2} + \cdots + w_{pn} = 1, \qquad (7)$$

where $R(i)$ is the average weighted ranking value for the object i, $r_{pn}(i)$ is the ranking of the object i from the perspective of pn, and w_{pn} is the weight for the perspective of pn. The w_{pn} can be customized by a specific task or a specific user, and the number of perspectives can also be specified.

Following the task and results shown in Table 7, 8 gives an illustrative example on how two different perspectives can be considered together and how the results can be merged. For simplification, here we give the weights for the publication number perspective and the coauthor number perspective the same value, namely, $w_{p1} = w_{p2} = 0.5.r_{cn}$ and r_{pn} represent the ranking from the coauthor number perspective and the publication perspective, respectively, and $R(i)$ represents the average weighted rank value from the two perspectives. From Table 8, we can conclude that for those which are ranked to the top from one perspective are not necessarily going to be ranked to the same position if we consider the ranking from the merging of different perspectives.

Table 9 A comparative study of the multi-level completeness strategy without and with a starting point (user name: John McCarthy)

Completeness	Authors (coauthor numbers) without a starting point	Authors (coauthor numbers) with a starting point
Level 1 $degree(n, P_{cn}) \geq 70$	Carl Kesselman (312)	Hans W. Guesgen (117) *
	Thomas S. Huang (271)	Carl Kesselman (312)
	Edward A. Fox (269)	Thomas S. Huang (271)
	Lei Wang (250)	Edward A. Fox (269)
	John Mylopoulos (245)	Lei Wang (250)
	Ewa Deelman (237)	John Mylopoulos (245)
	…	…
Level 2 $degree(n, P_{cn}) \in [30, 70)$	Claudio Moraga (69)	Virginia Dignum (69) *
	Virginia Dignum (69)	John McCarthy (65) *
	Ralph Grishman (69)	Aaron Sloman (36) *
	Biplav Srivastava (69)	Claudio Moraga (69)
	Ralph M. Weischedel (69)	Ralph Grishman (69)
	Andrew Lim (69)	Biplav Srivastava (69)
	…	…
…	…	…

5.4 Integration of different strategies

Although each of the proposed strategies is designed to meet one type of user needs, we would like to emphasize that the proposed strategies could be integrated together in order to provide users with refined results or satisfy more complex user needs.

We here give an illustrative example that integrates the multi-level completeness strategy and the starting point strategy together and provides to the user a more preferred list of reasoning results. Following the same task in the above sections, "John McCarthy" is taken as a concrete user name in a SP, a comparative list of partial results acquired by the multi-level completeness strategy without and with a starting point is provided in Table 9.

In Table 9, the levels are divided based on $degree(n, p_{cn})$ which is provided by the user to stop the query. The results in the left column are acquired only using the multi-level completeness strategy, and the results are just some regular author names distributed in different levels. While in the right column, results are from an integration of the starting point strategy and the multi-level completeness strategy. The coauthors are all persons whom the author should know, and their names serve as more "familiar results" compared to other names. These names serve as the user-related background information in the starting point and help users get more convenient results. Hence, for the right column, the coauthors[9] whom the user "John McCarthy" definitely knows (with "*" after the names) are ranked into the top ones in every level of the "Artificial Intelligence" author lists. Some users may prefer this kind of results, since advantages of two different strategies are integrated together. The user do not have to wait for the whole list of results if they are satisfied with a certain level of

---

[9] In this study, we represent the coauthor information for each author in an RDF file using the FOAF vocabulary "foaf:knows". The coauthor network RDF dataset, which was created based on the SwetoDBLP dataset, can be acquired from http://www.wici-lab.org/wici/dblp-sse. One can utilize this dataset to create a starting point for refining the reasoning process.

completeness, meanwhile, in each level, the user gets most familiar results first, which are relevant to user backgrounds and may be interesting for them.

The previous section only illustrates how the multi-level completeness strategy can be integrated with the starting point strategy. One can try to integrate other strategies together to satisfy various user needs. For example, the starting point strategy can also be integrated with the multi-level specificity strategy. Similar to the illustration in Table 9, in each levels of specificity, results can be ranked based on how close they are related with the starting point. The multi-level completeness strategy can also be integrated with the multi-level specificity strategy, in each level of specificity, the completeness of the results can be predicted. In our future study, we are going to investigate on how these different strategies can be better combined together to meet complex needs and provide better results.

6 Related work

This study is based on the framework of unifying search and reasoning proposed in Fensel and Harmelen [15]. The strategies introduced in this paper aim at providing some possible solutions for how the unification of search and reasoning can be implemented in a more user-oriented way from the viewpoint of granularity. They are developed based on many existing studies. We make some explanations on how they are related with and different from this study.

Expanding queries from user profiles has been studied in the field of information retrieval and Semantic Web search [8,13,29,31]. Many of them are based on user inputs of interests and cannot reflect the dynamic changing process through time. For the dynamic user interest modeling, one of the most important methods is to use a forgetting mechanism to model the interest shifting process. A linear gradual forgetting function has been used to model the shift of user interests and applied to user modeling in a recommender system [20]. The linear model is a simplification of the forgetting mechanism. In our study, a comparative study of interests retention using both an exponential law- and a power law-based models is provided. We also emphasized that the process of query extension is actually a refinement of search by reasoning.

The study of hierarchical searching has been studied from the knowledge organization perspectives. Mereology (a mathematical field of study that concentrate on the part-whole relationships) is used to generate hierarchical search results for user inspection [4]. More recently, hierarchical knowledge structures are used to supervise expert findings [45]. The study of reasoning with granularity starts from the logic approaches for granular computing [17,21,44], etc. Under the term of granular reasoning, it has also been studied from the perspectives of propositional reasoning [26], and granular space [36], etc. These studies concentrate on the logic foundations for reasoning under multi-granularity (mainly on zoom-in and zoom-out). Combining search results by reasoning has been proposed to deal with heterogeneous search results [11]. In this paper, we focus on how to unify the search and reasoning process from the viewpoint of granularity, namely, how to search for a good subset of the original dataset, and do reasoning on the selected dataset based on the idea of granularity. Besides the inspiration from granular computing [39,43], the strategies are also inspired from Cognitive Psychology (e.g. basic level) [30,34]. Further, we concentrate on how granularity-based strategies can help to effectively solve large-scale reasoning problems according to different user context and time constraints.

We also need to point out that although the strategies introduced in this paper are inspired by some basic strategies in granular computing, the granular structures, more

specifically granular knowledge structures that are mentioned in this paper are different from previous studies. In traditional models of granular computing, granules are organized hierarchically from larger grain sizes to smaller ones (or the other way around), and the granules in coarser levels contain the ones in finer levels [19,39]. In this study, although granules are still in a hierarchy, the granules do not contain each other. In the multi-level completeness strategy, granules are organized into different levels by the node degree under a perspective, granules with a higher value of $degree(n, p)$ do not contain those with lower values. In the multi-level specificity strategy, although the hierarchical knowledge structures of Artificial Intelligence have a typical granular structure (all the subtopics are covered under the terms one level coarser than them), the granular structure of the reasoning results based on this hierarchy is different from the granular structures studied previously [39,42], since the results that were got from the coarser levels cannot cover finer levels (the reason is that if the user does not have enough time, nodes in finer levels, such as authors of "Decision Trees" will not be selected for the reasoning task whether they are AI authors).

Variable precision logic is a major method for reasoning under time constraints, which provides two reasoning strategies, namely, variable certainty and variable specificity reasoning [23]. Concerning time constraint, given more time, a system with variable specificity can provide a more specific answer, while a system with variable certainty can provide a more certain answer [23]. Some strategies on unifying search and reasoning introduced in this paper, for example, the multi-level specificity strategy is inspired by variable specificity reasoning. The major difference is that: variable specificity reasoning uses "if-then-unless" rules, while multi-level specificity strategy uses a hierarchical knowledge structure to supervise the unification process of search and reasoning. In this paper, we did not investigate on the idea of variable certainty. Since it belongs to non-monotonic reasoning, the certainty will not necessarily go higher as more data are involved (since there might be contradictions [10] or inconsistency [18] on the facts, especially in the dynamic changing context of the Web). How it can be applied to a more user-centric environment still needs further investigations.

7 Conclusion and future work

In this paper, we provide several strategies for query refinement and processing realizing the user contexts and based on granularity. With user involvement, switching among different levels and views during the unification process of search and reasoning is the basic idea of this study. Through interaction between users and the system, the process is not fully automatic, but an interactive process.

The starting point strategy focuses on user-specific background (retained interests that are selected for constructing the starting point are based on proposed interests retention models) and the unification process is familiarity driven or novelty driven. The multi-level completeness strategy is with anytime behavior (i.e. it can be stopped at any given time) [33] and provides predictions of completeness for user judges when the user interacts with the system. In the multi-level completeness strategy, although the partial results may have low completeness, more important results have been searched out and ranked to the top ones for reasoning based on their higher values of $degree(n, p)$. The multi-level specificity strategy emphasizes on query processing with multiple levels of specificity and users can choose whether to go into more specific or more general levels. This strategy concentrates on the appropriate levels of specificity controlled by the knowledge hierarchy and does not get

into unnecessary levels. Hence, the time of problem-solving tasks is reduced. Furthermore, the multi-perspective strategy attempts to meet various user needs from multiple perspectives.

In this study, we acquire top 9 interests according to their values (such as retained interests values) for the starting point strategy. The reason is that, in Cognitive Psychology, research results proofed that the average human working memory can hold 7 ± 2 objects [24]. Although the interests are different from objects in working memory, we feel these two are to some extent related, but currently, there are no solid evidence to prove it. Hence, further studies are needed for how many interests should we keep for refining the query. In the current stage, semantic similarities of the interests have not been addressed into the interest retention models. This may affect the sequence of interests retention ranking, since it may be better to, in some way combine semantically very similar terms (such as synonyms) and then calculate the interest retentions. In the future work, we would like to calculate the semantic similarities of interesting terms so that more accurate retained interests can be acquired and better search constraints can be found. Since user needs are very related to the satisfaction of problem-solving results, as future work, we plan to provide a comparative study from the user perspective on the effects of multiple strategies mentioned in this paper. As introduced in Sect. 5.4, we also plan to have more deeper investigation on how these strategies can be combined together to produce better solutions. In order to further develop these strategies to meet the growing number of data sources, we will develop parallel, distributed implementations of these strategies in a knowledge grid architecture [9]. Currently, we just carried out a set of experiments in the context of query refinement and processing on large-scale scientific literatures. In the near future, we are going to investigate on how the proposed methods can be applied to wider ranges.

References

1. Aleman-Meza B, Hakimpour F, Arpinar IB, Sheth AP (2007) Swetodblp ontology of computer science publications. Web Semant Sci Serv Agents World Wide Web 5(3):151–155
2. Anderson JR, Schooler LJ (1991) Reflections of the environment in memory. Psychol Sci 2(6):396–408
3. Antoniou G, van Harmelen F (2008) A semantic web primer. 2. The MIT Press, Massachusetts
4. Arnold SE (2001) Rough sets, ants, and mereology: a new approach to knowledge management. Information World Review, submitted on August 2001
5. Barabási A (2002) Linked: the new science of networks. Perseus Publishing, Massachusetts
6. Bargiela A, Pedrycz W (2002) Granular computing: an introduction. 1. Kluwer Academic, Dordrecht
7. Berners-Lee T, Fischetti M (1999) Weaving the web: the original design and ultimate destiny of the world wide web by its inventor. Harper, SanFrancisco
8. Bhatia SK (1992) Selection of search terms based on user profile. In: Proceedings of the 1992 ACM/SIGAPP symposium on applied computing: technological challenges of the 1990's. ACM Press, Missouri, USA, pp 224–233
9. Cannataro M, Talia D (2003) The knowledge grid. Commun ACM 46(1):89–93
10. Carnielli WA, del Cerro LF, Lima-Marques M (1991) Contextual negations and reasoning with contradictions. In: Proceedings of the 12th international joint conference on artificial intelligence, pp 532–537

11. Ceri S (2009) Search computing. In: Proceedings of the 2009 IEEE international conference on data engineering. IEEE Press, pp 1–3
12. Collins AM, Quillian MR (1969) Retrieval time from semantic memory. J Verbal Learn Verbal Behav 8:240–247
13. Daoud M, Tamine-Lechani L, Boughanem M (2009) Towards a graph-based user profile modeling for a session-based personalized search. Knowl Inf Syst 21(3):365–398
14. Ebbinghaus H (1913) Memory: a contribution to experimental psychology Hermann Ebbinghaus. Teachers College, Columbia University, New York
15. Fensel D, van Harmelen F (2007) Unifying reasoning and search to web scale. IEEE Internet Computing 11(2):96, 94–95
16. Fensel D, van Harmelen F, Andersson B, Brennan P, Cunningham H, Valle ED, Fischer F, Huang Z, Kiryakov A, Lee TK, School L, Tresp V, Wesner S, Witbrock M, Zhong N (2008) Towards larkc: a platform for web-scale reasoning. In: Proceedings of the 2008 IEEE international conference on semantic computing. Washington, DC, USA, pp 524–529
17. Hobbs JR (1985) Granularity. In: Proceedings of the ninth international joint conference on artificial intelligence. Morgan Kaufmann, Los Angeles, USA, pp 432–435
18. Huang ZS, van Harmelen F, ten Teije A (2005) Reasoning with inconsistent ontologies. In: Proceedings of the 19th international joint conference on artificial intelligence. Edinburgh, UK, pp 454–459
19. Inuiguchi M, Hirano S, Tsumoto S (2003) Rough set theory and granular computing. 1. Springer, Berlin
20. Koychev I (2000) Gradual forgetting for adaptation to concept drift. In: Proceedings of ECAI 2000 workshop current issues in spatio-temporal reasoning. Berlin, Germany, pp 101–106
21. Liu Q, Wang QY (2005) Granular logic with closeness relation λ and its reasoning. In: Lecture notes in computer science, vol 3641, pp 709–717
22. Loftus GR (1985) Evaluating forgetting curves. J Exp Psychol Learn Mem Cogn 11:397–406
23. Michalski RS, Winston PH (1986) Variable precision logic. Artif Intell 29(2):121–146
24. Miller GA (1956) The magical number seven, plus or minus two: some limits on our capacity for processing information. Psychol Rev 101(2):343–352
25. Minsky M (2006) The emotion machine : commonsense thinking, artificial intelligence, and the future of the human mind. Simon & Schuster, New York
26. Murai T, Resconi G, Nakata M, Sato Y (2003) Granular reasoning using zooming in & out: Propositional reasoning. In: Lecture notes in artificial intelligence, vol 2639, pp 421–424
27. Myers JL, Well AD (2002) Research design and statistical analysis. 2. Routledge, London
28. Newell A, Rosenbloom PS (1981) Cognitive skills and their acquisition, chapter mechanism of skill acquisition and the law of practice. Lawrence Erlbaum Associates Inc, Hillsdale 1–55
29. Pretschner A, Gauch S (1999) Ontology based personalized search. In: Proceedings of the 11th IEEE international conference on tools with artificial intelligence, pp 391–398
30. Rogers T, Patterson K (2007) Object categorization: reversals and explanations of the basic-level advantage. J Exp Psychol Gen 136(3):451–469
31. Tamine-Lechani L, Boughanem M, Daoud M (2009) Evaluation of contextual information retrieval effectiveness: overview of issues and research. Knowl Inf Syst, published online, July 2009
32. Triola MF (2005) Elementary statistics, using the graphing calculator: for the TI-83/84 plus. Pearson Education
33. Vanderveen K, Ramamoorthy C (1997) Anytime reasoning in first-order logic. In: Proceedings of the 9th international conference on tools with artificial intelligence. IEEE Press, Washington, DC, USA, pp 142–148
34. Wickelgren WA (1976) Handbook of learning and cognitive processes: vol 6: linguistic functions in cognitive theory, chapter memory storage dynamics. Lawrence Erlbaum Associates, Hillsdale 321–361
35. Wisniewski EJ, Murphy GL (1989) Superordinate and basic category names in discourse: a textual analysis. Discourse Processing 12:245–261
36. Yan L, Liu Q (2008) Researches on granular reasoning based on granular space. In: Proceedings of the 2008 international conference on granular computing, vol 1. IEEE Press, Honululu, pp 706–711
37. Yao YY (2005) Perspectives of granular computing. In: Proceedings of 2005 IEEE international conference on granular computing, vol 1. Beijing, China, pp 85–90
38. Yao YY (2007) The art of granular computing. Lect Notes Artif Intell 4585:101–112
39. Yao YY (2008) Handbook of granular computing, chapter A unified framework of granular computing. Wiley, New York, pp 401–410
40. Zeng Y, Wang Y, Huang ZS, Zhong N (2009) Unifying web-scale search and reasoning from the viewpoint of granularity. Lect Notes Comput Sci 5820:418–429
41. Zeng Y, Yao YY, Zhong N (2009) Dblp-sse: a dblp search support engine. In: Proceedings of the 2009 IEEE/WIC/ACM international conference on web intelligence. IEEE Press, pp 626–630

42. Zeng Y, Zhong N (2008) On granular knowledge structures. In: Proceedings of the first international conference on advanced intelligence. Posts & Telecom Press, Beijing, China, pp 28–33
43. Zhang B, Zhang L (1992) Theory and applications of problem solving. 1. Elsevier Science Inc, Amsterdam
44. Zhou B, Yao YY (2008) A logic approach to granular computing. Int J Cogn Inf Nat Intell 2(2):63–79
45. Zhu JH, Huang XJ, Song DW, Rüger S (2010) Integrating multiple document features in language models for expert finding. Knowl Inf Syst 23(1):29–54

Author Biographies

Yi Zeng is currently a Ph.D. student at the International WIC Institute, Beijing University of Technology, Beijing, China. His research interests include Web Intelligence, Knowledge representation and reasoning, and Semantic Web. He received a B.E. degree from Beijing University of Technology, Beijing, China, in 2004. He used to visit University of Regina, Canada, from May to July in 2007, and Vrije University Amsterdam, the Netherlands, from May to June in 2009, respectively, doing research projects related to knowledge retrieval and reasoning.

Ning Zhong is currently the head of the Knowledge Information Systems Laboratory and a Professor in the Department of Life Science and Informatics at Maebashi Institute of Technology, Japan. He is also a director and an adjunct professor in the International WIC Institute, Beijing University of Technology. He has conducted research in the areas of knowledge discovery and data mining, rough sets and granular-soft computing, Web intelligence, intelligent agents, brain informatics, and knowledge information systems, with more than 200 journal and conference publications and 20 books. He is the Editor-in-Chief of Web Intelligence and Agent Systems and serves as associate editor/editorial board for several international journals and book series. He is the co-chair of Web Intelligence Consortium (WIC), chair of IEEE-CIS Task Force on Brain Informatics. He has served as chair of the IEEE-CS Technical Committee on Intelligent Informatics, ICDM'02 (conference chair), ICDM'06 (program chair), WI-IAT'03 (conference chair), WI-IAT'04 (program chair), IJCAI'03 (advisory committee member), and Brain Informatics 2009 (program chair). He was awarded IEEE TCII/ICDM Outstanding Service Award in 2004, and Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD) Most Influential Paper Award (1999–2008).

Yan Wang received a B.E. degree of Computer Science from Zhengzhou University of Light Industry, Zhengzhou, China, in 2000. She is currently a Ph.D. Student at the International WIC Institute, Beijing University of Technology, Beijing, China. Her research interests include Web Intelligence, Semantic Web, Knowledge representation and reasoning. She is a member of the LarKC project funded by the European Commission under Contract Number FP7-215535.

Yulin Qin is a Distinguished Professor at the International WIC Institute, Beijing University of Technology, Beijing, China, and Senior Research Psychologist at Carnegie Mellon University, Pittsburgh, PA, USA. He received his M.E. degree in Computer Science and Engineering in 1982 at Beijing Institute of Aeronautics and Astronautic, Beijing, China, and Ph.D. degree 1992 in Cognitive Psychology at Carnegie Mellon University, Pittsburgh, PA, USA. He learnt cognitive neuroscience including using multiple channel single-cell recording technology to explore the functions of spatial representation and memory consolidation in hippocampal structure of the rats during his postdoc training periods. He is currently interested in the neural basis of human cognitive architecture, as well as the web intelligence systems inspired by cognitive neuroscience.

Zhisheng Huang is a senior researcher in the group of Knowledge Representation and Reasoning at Vrije University of Amsterdam, the Netherlands. He is also an adjunct professor of Faculty of Computer Science, Southeast University, and an adjunct professor of Jiangsu University of Science and Technology, China. He received his B.E. and M.Sc. in Computer Science from Harbin Engineering University and received Ph.D. in logics and computer science in 1994 from University of Amsterdam, the Netherlands. He has published about 100 papers in logics, artificial intelligence, multimedia, and the Semantic Web. He serves as a program committee member for over forty international workshops/conferences. He co-chaired several workshop/conferences, which include the 2009 Chinese Semantic Web Conference (CSWS2009) and the First Asian Workshop on Scalable Semantic Data Processing (AS2DP2009).

Haiyan Zhou is an assistant professor of the International WIC Institute of Beijing University of Technology. She received her Bachelor of Psychology in Hunan Normal University and her Master of Cognitive Psychology in Beijing Normal University, China. In 2007, she received her Ph.D. in Cognitive Neuroscience from the National Key Laboratory for Cognitive Neuroscience and Learning, Beijing Normal University. Her major research interests are brain informatics and its application in Web intelligence. She has published about 10 papers about human cognitive neuroscience.

Yiyu Yao is a professor of computer science in the Department of Computer Science, University of Regina, Regina, Saskatchewan, Canada. His research interests include information retrieval, rough sets, interval sets, granular computing, Web intelligence, data mining, and fuzzy sets. His publications cover various topics on modeling information retrieval system based on user preferences, information retrieval support systems, triarchic theory of granular computing, generalized rough sets, the foundations of data mining, and many more.

Frank van Harmelen is a full professor in Knowledge Representation and Reasoning at VU University Amsterdam, the Netherlands. After studying Mathematics and Computer Science in Amsterdam, he obtained his Ph.D. from the University of Edinburgh (Department of AI) for his research on meta-level reasoning. He is the scientific director of the EU 7th framework project LarKC, aiming to build the Large Knowledge Collider. He was one of the designers of OWL, the W3C standard Web Ontology Language. He is the scientific advisor of Aduna, one of the earliest companies in the Semantic Web arena, and developers of the Sesame RDF storage and retrieval engine. He has published over 100 papers, many of them in leading journals and conferences, and many of them highly cited. One of his five books is the Semantic Web Primer, the first text book on Semantic Web technology (now deployed in university courses across the world, with translations in Spanish, Japanese, Chinese and Korean).