Inf Retrieval (2014) 17:52–73
DOI 10.1007/s10791-013-9228-1

Leveraging integrated information to extract query subtopics for search result diversification

Wei Zheng · Hui Fang · Conglei Yao · Min Wang

Received: 8 October 2012 / Accepted: 21 February 2013 / Published online: 4 July 2013

Abstract Search result diversification aims to diversify search results to cover different *query subtopics*, i.e., pieces of relevant information. The state of the art diversification methods often explicitly model the diversity based on query subtopics, and their performance is closely related to the quality of subtopics. Most existing studies extracted query subtopics only from the unstructured data such as document collections. However, there exists a huge amount of information from structured data, which complements the information from the unstructured data. The structured data can provide valuable information about domain knowledge, but is currently under-utilized. In this article, we study how to leverage the integrated information from both structured and unstructured data to extract high quality subtopics for search result diversification. We first discuss how to extract subtopics from structured data. We then propose three methods to integrate structured and unstructured data. Specifically, the first method uses the structured data to guide the subtopic extraction from unstructured data, the second one uses the unstructured data to guide the extraction, and the last one first extracts the subtopics separately from two data sources and then combines those subtopics. Experimental results in both Enterprise and Web search domains show that the proposed methods are effective in extracting high quality subtopics from the integrated information, which can lead to better diversification performance.

W. Zheng (✉) · H. Fang
University of Delaware, 209 Evans Hall, Newark, DE, USA
e-mail: zwwaynee@gmail.com

H. Fang
e-mail: hfang@udel.edu

C. Yao
Tencent, Beijing, China
e-mail: ycl.pku@gmail.com

M. Wang
HP Labs China, Beijing, China
e-mail: min.wang6@hp.com

Keywords Web search · Enterprise search · Diversification · Query subtopics · Structured data · Unstructured data

1 Introduction

Top ranked results of traditional retrieval models often contain relevant yet redundant information and may not satisfy all possible information needs that are hidden behind a keyword query. Since search queries could be ambiguous or under-specified, result diversification is an important technique to improve user search experience through maximizing the coverage of relevant information while minimizing the redundancy in search results (Agrawal et al. 2009; Carbonell and Goldstein 1998; Carterette and Chandar 2009; Dang et al. 2011; Demidova et al. 2010; Clarke et al. 2009a; Lubell-Doughtie and Hofmann 2011; Santos et al. 2010a; Radlinski and Dumais 2006; Santos et al. 2010b, c). The state of the art diversification methods often diversify search results explicitly based on query subtopics, and the goal is to maximize the coverage of query subtopics in the top ranked documents (Agrawal et al. 2009; Santos et al. 2010a). Clearly, the diversification performance is closely related to the quality of subtopics.

Many search domains contain not only an *unstructured* document collection that needs to be searched but also a companion set of *structured* data that can provide valuable information about the domain knowledge of retrieval. For example, in Web search domain, the data include unstructured web pages as well as structured information such as those from Semantic Web. Enterprise search is another important domain with both structured and unstructured data (Hawking 2004). Intuitively, these two types of information are complementary to each other, and each of them has its own unique advantages in providing valuable information for subtopic extraction. Specifically, unstructured documents can provide collection-specific subtopics while structured data are more effective in providing high-quality yet domain-dependent subtopics. Existing query subtopic extraction methods focused on extracting subtopics from either document collections (Balog et al. 2009a; Bi et al. 2009; Carterette and Chandar 2009; Dang et al. 2011; Dou et al. 2009; He et al. 2010; Li et al. 2009; Lubell-Doughtie and Hofmann 2011; Zheng et al. 2011b) or taxonomies (Agrawal et al. 2009; Hauff and Hiemstra 2009; Santos et al. 2010b). It remains under-studied how to leverage both types of data to extract query subtopics with better quality.

In this paper, we study the problem of extracting query subtopics from both structured and unstructured data. The structured data, e.g., databases, often contain valuable domain knowledge such as concept hierarchy. The concept hierarchy in the structured data, i.e., databases, can provide useful clues on the extraction of query subtopics. In subtopic extraction, it is often difficult to determine whether we should select general query subtopics (e.g., java coffee vs. java programming) or more specific ones (e.g., java programming update vs. java programming tutorial). To solve the problem, we propose to dynamically determine the subtopic level of a query by selecting nodes that are more relevant to the query from the concept hierarchy as query subtopics. As a result, when the selected nodes are at higher-level in the concept hierarchy, we can infer that the query subtopics need to be more general and vice versa.

However, since we search for diversified information from the *unstructured* data, using subtopics extracted from *structured* data could potentially have the problem of vocabulary

mismatch and may not be effective in diversification. To overcome this limitation, we then propose three methods to integrate structured and unstructured data to extract high quality subtopics that are effective in diversifying documents. The first method uses the structured data to guide the subtopic extraction from the unstructured data. The second one uses the unstructured data to guide the subtopic extraction from the structured data. And the third one combines subtopics that are extracted separately from the unstructured data and structured data.

We then use a state of the art diversification method to diversify the documents based on the extracted subtopics. We evaluate the effectiveness of the proposed methods through the diversification performance. We first conduct the experiments over a real-world enterprise data collection containing both unstructured documents and structured relational databases. Results show that all of the three proposed subtopic integration methods can extract high quality subtopics and improve the diversification performance. To validate the effectiveness of the proposed methods in Web search, we conduct experiments over standard TREC collections and use the Open Directory Project (ODP) as the complementary structured data. Experimental results over this collection are similar to the ones in Enterprise search, i.e., it is more effective to diversify search results using the query subtopics that integrate structured and unstructured data. Moreover, we find that the third integration method combining subtopics from structured and unstructured data is the most robust method that performs well on both collections.

2 Overview of subtopic-based diversification methods

The goal of search result diversification is to return a list of documents which are not only relevant to a query but also cover different subtopics of the query. Here a query subtopic refers to a representative information need associated with the query (Clarke et al. 2009a). Most existing diversification methods, such as IA-Select (Agrawal et al. 2009), xQuAD (Santos et al. 2010a, c) and SQR (Zheng et al. 2012), are subtopics-based.

Given a query, these studies would first identify and extract subtopics for the query, and then retrieve a list of documents with the maximal relevance and diversity score based on these query subtopics. Finding the optimal solution for this problem has been proved to be NP-hard, so a greedy algorithm is often used (Agrawal et al. 2009). Specifically, it starts with an empty document set, and then iteratively selects a local optimal document that can maximize both relevance and diversity of the new document list after adding the selected document to the previously selected list. The relevance score of a document list can be computed using any existing retrieval function. Existing diversification methods mainly differ in how to compute the diversity score of a document list based on the query, its subtopics and the documents in the list (Agrawal et al. 2009; Carterette and Chandar 2009; Santos et al. 2010a; Radlinski and Dumais 2006; Zheng et al. 2012).

In this paper, we use SQR, i.e., one of the recently proposed diversification methods, to diversify search results. This method is chosen because of its effectiveness in result diversification (Zheng et al. 2012). We now provide more details about how this method models the diversity score.

SQR is derived from a coverage-based optimization framework, which models diversity based on the coverage of query and subtopics. In particular, the diversity score of a document list D with respect to query q can be computed as the summation of the weighted coverage for all the query subtopics, which is shown as follows:

$$div(D,q) = \sum_{s \in S(q)} weight(s,q) \times cov(D,s) = \sum_{s \in S(q)} weight(s,q) \times \left(1 - \left(1 - \sum_{d \in D} cov(d,s)\right)^2\right), \quad (1)$$

where $S(q)$ is the subtopic set of query q and $weight(s, q)$ measures the weight of the subtopic s in the query q. $cov(D, s)$ measures the coverage of a specific subtopic s in D and is computed using a square-loss function based on the coverage of s in individual documents. Note that $weight(s, q)$ and $cov(s, d)$ can be estimated based on the similarity function of the two variables using any existing retrieval functions.

Clearly, the diversification performance of is closely related to the quality of query subtopics, i.e., $S(q)$.

3 Subtopic extraction from individual sources

In this section, we briefly describe how to extract subtopics from unstructured information, and propose a method to extract subtopics from structured information.

3.1 Subtopic extraction from unstructured information

One of the commonly used methods is to automatically extract subtopics of the query from document collections using topic modeling methods such as Probabilistic Latent Semantic Analysis (PLSA) (Lubell-Doughtie and Hofmann 2011). Specifically, the system would first retrieve a set of documents based on only relevance, and then use PLSA to generate M document clusters. Every term will be assigned to one of the clusters that is the most likely one to generate the term, and a cluster will be regarded as a query subtopic. Since terms in these subtopics are extracted directly from documents, they are effective to distinguish documents covering different pieces of relevant information.

3.2 Subtopic extraction from structured information

One limitation of using subtopics extracted from unstructured information is that they often contains lots of noisy terms. On the contrary, structured information is more accurate and contains high quality information. Specifically, the schema of relational databases can provide the meaning of the data and the domain knowledge. For example, the relations among different tables reveal a multi-level concept hierarchy of the data in the databases. Figure 1 shows an example of the database and its corresponding concept hierarchy. We join all the tables in a relational database into one table based on the foreign and primary keys in these tables. The first row in the joined table describes the schema information, which can be used to infer the concept hierarchy in the database. For example, one product type has multiple marketing categories and one marketing category has multiple product series, etc. Therefore, the levels of the concept hierarchy are *product type*, *marketing category*, *marketing subcategory*, *product series* and *product name*.

The challenge of extracting subtopics from a concept hierarchy is to decide which levels of nodes from the concept hierarchy should be selected as query subtopics. A simple strategy used by existing work is to pre-select a level from the concept hierarchy, and then

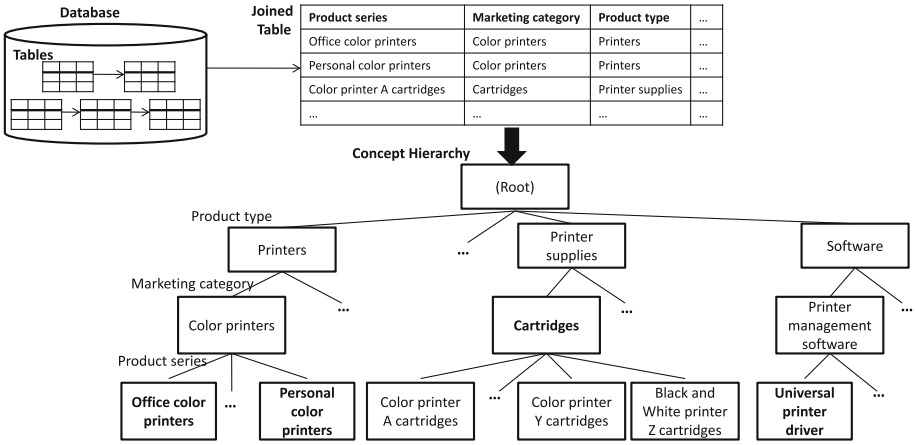

Fig. 1 A sample of the concept hierarchy constructed from the relational databases and the corresponding schema of each level. Highlighted nodes are the ideal set of subtopics of the query *"color printer"*

choose the nodes from that level as query subtopics to avoid generating overlapped subtopics (Agrawal et al. 2009; Lubell-Doughtie and Hofmann 2011; Hauff and Hiemstra 2009; Santos et al. 2010b). The nodes selected from the upper level will cover more general information while nodes selected from the lower level contain more specific information. However, it is difficult to decide an appropriate level that works well for all the queries since some queries could be more specific than others. Moreover, all the subtopics of a query do not need to be at the same level. For example, let us consider the concept hierarchy shown in Fig. 1 and query "*color printer*". A good choice of the nodes as subtopics is {"*office color printer*", "*personal color printer*", "*cartridges*", "*universal printer driver*"}. The node "*cartridges*" is on the third level while the other nodes are on the fourth level. On one hand, if we only select nodes from the third level, the subtopics will lose more specific information, i.e., *office* and *personal*. On the other hand, if we select nodes from only the fourth level, the subtopics may contain too many specific subtopics on different cartridges.

To overcome these limitations, we formulate the problem of extracting subtopics from a concept hierarchy as the one that selecting a set of nodes that are most relevant to the query. But how to compute the relevance score of a node? Recall that a node has an "is-a" relationship with its ancestor nodes in the concept hierarchy, which means the descendants of a node contain more specific information about the node. Thus, we propose to compute the relevance score of a node based on not only the query similarity of the node itself but also those of its descendants. Formally, the relevance score of node n with respect to query q is computed as follows:

$$rel(n,q) = \frac{\sum_{n_i \in N} sim(n_i, q)}{|N|^{\gamma}}, \tag{2}$$

where N is the set of nodes contained in the sub-tree rooted at n and includes node n as well as all of its descendants. $sim(n_i, q)$ is the similarity between the node n_i and the query, and we will discuss how to compute it later. γ is the parameter to adjust the impact of the sub-tree size on the relevance score. The value of γ can be any positive real numbers. When γ is 0, Eq. (2) is the sum of the scores of the nodes in N. In this case, the node n in the upper

level would have higher score than its descendant nodes in the lower level when $rel(n, q)$ is larger than 0. When γ is 1, Eq. (2) is the average score of the nodes in N (Zheng et al. 2011a). Thus, the node n will have lower score than its descendants if the similarity between n and q is smaller than the similarity between its descendants in a lower level and the query. It is clear that when the value of γ is larger, it is more likely to select nodes from lower levels.

Given a query, we can then select M nodes with the highest relevance scores from the concept hierarchy based on the above relevance function. Since one desirable property of query subtopics is that they should cover non-overlapping information about the query, we refine the process by iteratively selecting the most relevant node that is neither the ancestor nor descendant of previously selected nodes:

$$n^* = \arg \max_{n \in N_c \setminus ancdes(S)} rel(n, q), \tag{3}$$

where N_c is set of all the nodes from the concept hierarchy, S is the set of previously selected nodes and $ancdes(S)$ is the set of ancestor and descendant nodes of S.

Each selected node can then be regarded as a query subtopic. However, the description of each node is often short, and using the short description of a node as a query subtopic may not work well because of the possible vocabulary gap with the documents. Thus, we use not only the description of a selected node but also those from its ancestor nodes as the query subtopic. Ancestor nodes are chosen over descendant nodes because we want to avoid generating over-specific query subtopics.

One remaining challenge is how to compute the similarity between a node and a query, i.e., $sim(n_i, q)$ as shown in Eq. (2), based on the terms from n_i and q. Exact term matching methods unlikely work well because the query and nodes are often short and contain only a few terms. A node (e.g., "*cartridge*") related to a query (e.g., "*printer*") should have a high similarity with the query even if their terms do not match. To capture such semantic similarity, we follow existing studies and compute the similarity between query q and node n based on the co-occurrence information of their terms: (Fang and Zhai 2006; van Rijsbergen 1979):

$$sim(n_i, q) = \frac{\sum_{t_i \in n_i} \sum_{q_j \in q} sim(t_i, q_j)}{|n_i| \times |q|}, \tag{4}$$

where t_i a term from the description of node n_i, q_j is a query term from q, $|n_i|$ is the number of terms in n_i and $|q|$ is the number of query terms. $sim(t_i, q_j)$ is the normalized weighted mutual information value of the two terms, and the mutual information is computed over a working set with both documents relevant to the query and randomly selected ones from the collection (Fang and Zhai 2006).

The proposed subtopic extraction method is described in Algorithm 1. Lines 2–5 describe the step to compute relevance score of each node. Lines 7–22 describe the step to select non-overlapping nodes as subtopic candidates. The main advantage of the method is that it can dynamically select nodes from different levels.

4 Subtopic integration

Subtopics extracted from *unstructured* information contain terms that are effective in distinguishing documents covering different subtopics while they also contain lots of noisy

terms. On the contrary, subtopics extracted from *structured* information often contain high quality terms while there might be vocabulary gaps between the subtopic terms and retrieved documents. Thus, it is clear that the subtopics extracted from these two types of information are complementary. The subtopics could be more effective if they use both unstructured information and structured information.

ALGORITHM 1: Subtopic Extraction in the Structured Data

Input: Query (i.e.,q) , the query similarity score for node n (i.e., $sim(n, q)$), the node set from the concept hierarchy (i.e., N_c), the number of subtopics (i.e.,M, where $M \leq |N|$), and the parameter in the relevance function (i.e., γ).

Output: The subtopic set S_d where $|S_d| = M$

/*Step1: compute relevance score of each node*/
for $n \in N_c$ **do**
  $N_i = descendants(n) \cup n$
  $rel(n, q) = \frac{\sum_{n_i \in N_i} sim(n_i, q)}{|N_i|^{\gamma}}$
end for
/*Step2: select important nodes as subtopics*/
$S_d = \emptyset$
$N' = N_c$
while $|S_d| < M$ **do**
  $n^* = \arg\max_{n \in N'} rel(n, q)$
  $overlap = 0$
  for $s_d \in S_d$ **do**
    if $n^* \in (n \cup ancestors(s_d) \cup descendants(s_d))$ **then**
      $overlap = 1$
    end if
  end for
  if $overlap == 0$ **then**
    $S_d = S_d \cup \{n^* \cup ancestors(n^*)\}$
  end if
  $N' = N' \setminus (n^* \cup ancestors(n^*) \cup descendants(n^*))$
end while
return S_d

We therefore systematically propose and study three methods that integrate structured and unstructured data to discover high-quality query subtopics that are effective in diversifying documents. The first method extracts subtopics from the structured data and uses these subtopics to guide the subtopic extraction in the unstructured data. The second method uses the unstructured data, documents, to select subtopics from the structured data. The third method extracts subtopics separately from structured and unstructured data, and then combines these subtopics.

4.1 Subtopic integration guided by structured data (*StruGuide*)

The limitation of using only unstructured data, i.e., documents, for subtopic extraction is that documents may contain lots of off-topic terms, which could lead to noisy subtopics. To overcome this limitation, we propose to use the high-quality information from databases to guide the selection of subtopics from the documents. The basic idea is to use databases to extract high-quality subtopics and then use documents to bridge the vocabulary gap between documents and subtopics. Specifically, we first extract subtopics from the structured information using the method described in Sect. 3.2, and then use these subtopics as

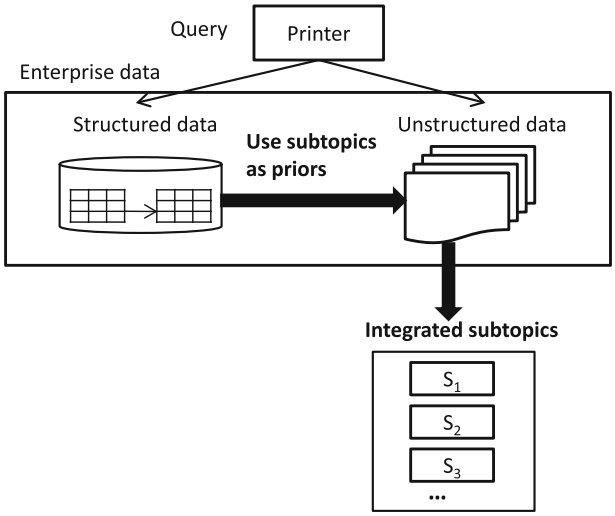

Fig. 2 The subtopic integration method guided by the structured data (**StruGuide**)

the prior of PLSA method (Mei et al. 2007) to generate the integrated subtopics. The process is described in Fig. 2.

4.2 Subtopic integration guided by unstructured data (*UnstruGuide*)

One limitation of using only structured information, databases, to extract subtopics is that queries are too short and may not be effective to select different subtopics from the concept hierarchy. One solution is to use relevant documents of a query to represent the query and select the relevant nodes. Following the same assumption made in the studies on pseudo feedback, we assume that top-ranked R documents are relevant and refer them to as pseudo relevant documents. R is set to 60 as recommended in the previous study (Zheng et al. 2012), and the documents are ranked based on their relevance scores with respect to the query.

The integration process is shown as in Fig. 3. We use each top-ranked document to select the most relevant node from the concept hierarchy and use the node as one of the query subtopics. The relevance of a node to the document is computed using Eq. (2). In fact, this method can be regarded as using *unstructured* information to guide the subtopic extraction from *structured* information. One advantage of the method is that we do not need to specify the number of subtopics because the number of subtopics is automatically determined by the top-ranked documents. Note that a single node may be selected by multiple documents, so the number of subtopics is often much smaller than the number of top-ranked documents.

4.3 Subtopic integration combining subtopics of unstructured and structured data (*Combine*)

The subtopics extracted from databases are not effective in diversifying documents because these subtopics may have a vocabulary gap with the documents, while subtopics extracted from documents may have a lot of noisy terms and cover overlapped information with each other. We therefore propose the third method that combines the subtopics extracted from the database and the documents, as shown in Fig. 4.

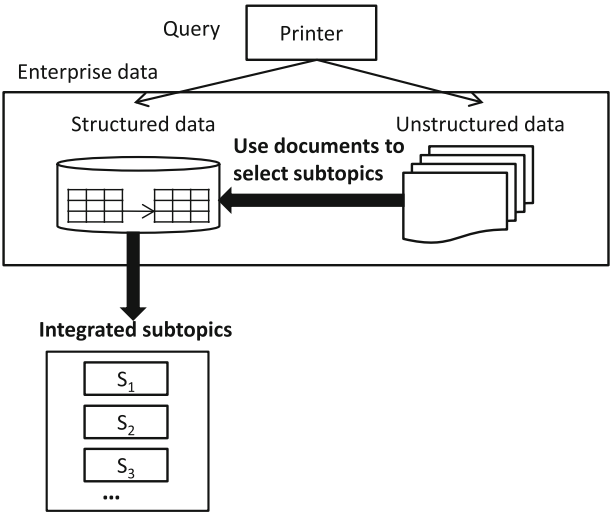

Fig. 3 The subtopic integration method guided by the unstructured data (**UnstruGuide**)

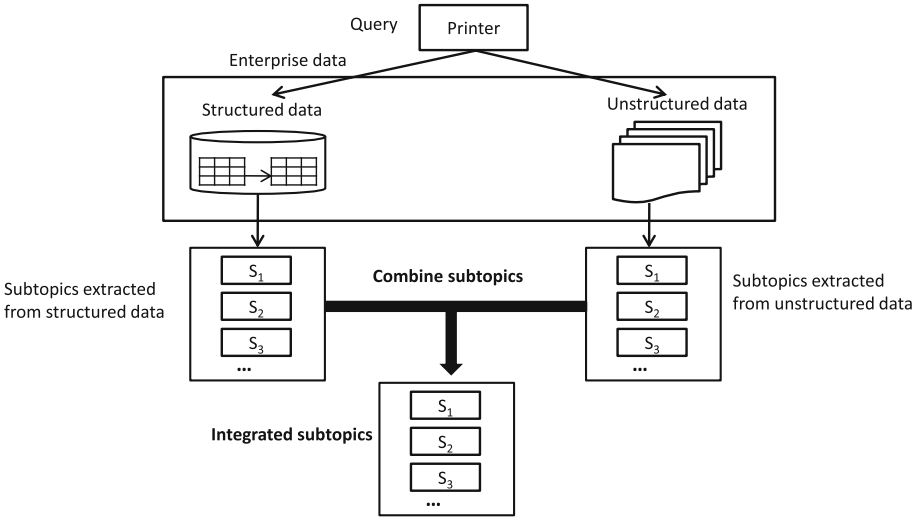

Fig. 4 The subtopic integration method combining subtopics extracted from the structured and unstructured data (**Combine**)

For each query, we first separately extract M subtopics from the database (denoted as DB subtopics) and M subtopics from the documents (denoted as DOC subtopics). For each DOC subtopic, we then connect it with the most similar DB subtopic. Each connected pair will be used to generate an integrated subtopic. The terms of an integrated subtopic are selected from its corresponding DOC subtopic by filtering out terms that are not semantically similar to the connected DB subtopic (Fig. 5).

We now describe the details of our method in Algorithm 2. The first step is to connect a subtopic extracted from retrieved documents with a subtopic extracted from databases. For each DOC subtopic, we find the most similar DB subtopic based on their semantic

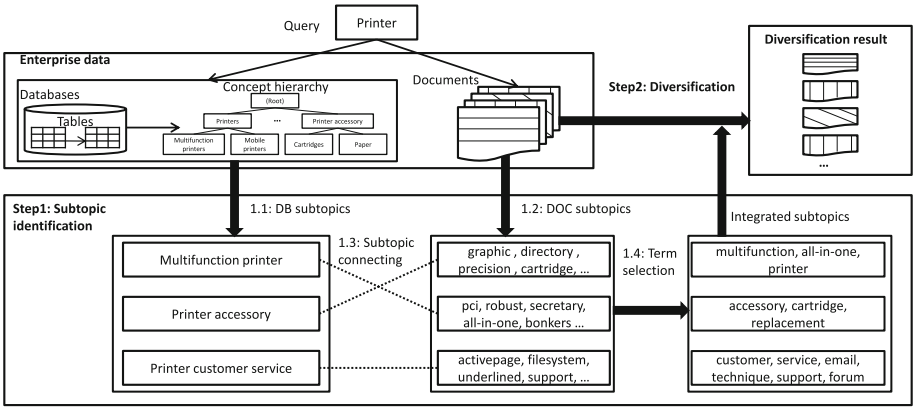

Fig. 5 An example to show the process of subtopic integration method *Combine*

similarity computed by Eq. (4), i.e., lines 2–12 in Algorithm 2. We then filter out noisy terms in DOC subtopic and keep K terms as the terms in the new integrated subtopic. The term selection criteria is that we select the terms that have the highest relevance score with the corresponding DB subtopic, as described in lines 14–24 in Algorithm 2.

ALGORITHM 2: Subtopic Integration Combining Subtopics Extracted from Structured and Unstructured Data

Input: A query q, a set of DB subtopic S_d, a set of DOC subtopics C, i.e., PLSA clusters, term set T, the cluster of each term $cluster(t)$, semantic similarity between the cluster and subtopic $rel(c_i, s_d)$ computed using Equation (4), number of subtopic terms K.

Output: The integrated subtopic set S where $|S| = M$.

1: /*Assigning database subtopics to clusters*/
2: $S'_d = S_d$
3: **for** $c_i \in C$ **do**
4: $s_{c_i} = \arg\max_{s \in S'_d} sim(c_i, s)$
5: $S'_d = S'_d \setminus \{s_{c_i}\}$
6: $T_{c_i} = \emptyset$ /*term set in the cluster*/
7: **for** $t \in T$ **do**
8: **if** $cluster(t) == c_i$ **then**
9: $T_{c_i} = T_{c_i} \cup t$
10: **end if**
11: **end for**
12: **end for**
13: /*Filtering out noisy terms from each cluster based on their similarity with database subtopics*/
14: $S = \emptyset$
15: **for** $c_i \in C$ **do**
16: $s^* = \emptyset$ /*remaining terms in integrated subtopic*/
17: $T'_{c_i} = T_{c_i}$
18: **while** $|s^*| < K$ *and* $|s^*| < |T_{c_i}|$ **do**
19: $t^* = \arg\max_{t \in T'_{c_i}} sim(t, s_{c_i})$
20: $s^* = s^* \cup t^*$
21: $T'_{c_i} = T'_{c_i} \setminus t^*$
22: **end while**
23: $S = S \cup \{s^*\}$
24: **end for**
25: **return** S

4.4 Discussions

The proposed integration methods use different methods to integrate information. *StruGuide* method enriches the DB subtopics by adding similar terms from documents. *UnstruGuide* method uses documents to guide the selection of subtopics from the database. *Combine* method uses databases to filter out noisy terms in the DOC subtopics.

Comparing the three methods, we make the following hypothesis. First, *UnstruGuide* is the one that would be most affected by the quality of the structured data. Since the subtopics have to come from the nodes of the concept hierarchy, the quality of the subtopics would be low when the information from the concept hierarchy does not complement search results from the unstructured data well. Second, *StruGuide* is the one that would be least affected by the quality of the structured data. Although the structured data is used as priors, the subtopics are actually the clusters of the documents, which would lead to smaller vocabulary gaps. Finally, *Combine* method is expected to work better than the other two since it can fully utilize the information from both sources.

The above integration methods assume that the subtopics extracted from either the structured or unstructured data are useful for diversification. However, the assumption might not hold when the structured data do not contain enough information about the query. Let us consider an example query "HP printer", and we want to search for relevant information from a document collection provided by HP. Assuming that we have a concept hierarchy extracted from IBM databases. Based on the proposed method, we might be able to extract query subtopics from this concept hierarchy, but they are clearly not very related to the query. As a result, these subtopics would not be effective to diversify search results.

It is clear that whether a query can benefit from the proposed integration methods depends on whether the structured data contain enough information about the query. This suggests that we should apply the proposed integration methods only to those queries that have enough relevant information in the structured data. But how can we predict the quality of structured data to a query?

We propose to make the prediction based on the average relevance score of the subtopics extracted from the structured data with respect to the query. The intuition is that the subtopics selected from the structured data should be more relevant to the query if the concept hierarchy contains more useful information about the query. The relevance score of each subtopic is computed using Eq. (2). We then compute the average relevance score over all the subtopics. When the average score is larger than a threshold, we use the integrated subtopics using both structured and unstructured data. When it is smaller, we only use the subtopics extracted from the documents, i.e., DOC subtopics extracted using PLSA method.

5 Experiments

We evaluate the effectiveness of the proposed subtopic extraction methods based on the diversification performance. In particular, we first use the proposed methods to extract query subtopics, apply the SQR method (described in Sect. 2) to diversify search results, and then evaluate the diversification performance in both Enterprise and Web search domains.

5.1 Effectiveness in search of an enterprise web

5.1.1 Experiment design

We first evaluate the proposed subtopic integration methods over a real world enterprise collection gotten from HP company. The data set includes both *unstructured data*, which consist of 477,800 web pages crawled from the web site of HP company, and *structured data*, which includes 25 relational databases of HP. Both the structured data and unstructured data are used for generating subtopics using the proposed methods. The unstructured data are then used for searching based on the extracted subtopics.

To evaluate the diversification performance, we select queries from a query log with 1,060,792 queries submitted to the enterprise search engine of HP during July 1st, 2010 and July 7th, 2010. One possible strategy of selecting queries is to randomly take samples from the query log. However, it may include rare queries which are less useful to evaluate the effectiveness of diversification. We therefore follow the similar procedure as that used in the diversity task of TREC Web track (Clarke et al. 2009a), and construct a query set with 50 queries that represent popular while different information needs. We first find popular query terms, use them to group similar queries, and then select representative information needs from different query groups. Specifically, the queries are selected from the query log in the following steps: (1) rank all the terms based on their frequencies in the query log; (2) remove stopwords and keep 100 most popular and meaningful terms which represent popular user needs; (3) rank all the queries in the query log based on the number of selected popular terms contained in the queries; and (4) manually select 50 queries from the top-ranked queries to cover different product information in the database.

The average number of terms per query for the query set is 3.66. Since the queries could be under-specified or ambiguous, the diversification is expected to improve user experience through maximizing the coverage of different query subtopics. For example, query "printer" contains different subtopics such as different types of printers, printer accessories, printer software and supporting information. The diversified search results would cover information relevant to all the subtopics.

Following the processes in TREC (Clarke et al. 2009a) and NTCIR (Song et al. 2011; Sakai and Song 2012), the human assessors constructed the real subtopics of the queries based on the representative information needs. Seven human assessors were hired to identify query subtopics and make relevance judgments. For each query, one of the assessors would first collect as much relevant information about the query as possible based on the related queries in the log, top ranked search results from the enterprise search engine and Web search engines. The assessors then created a set of query subtopics that can cover different representative information needs of the query. The average number of subtopics per query is 4.12, which is close to the number from TREC diversity collections, i.e., 4.61 on TREC 2009 and 4.36 on TREC 2010. The assessors also provided relevance judgments for documents with respect to each query subtopic. They evaluated the top ranked 200 documents in each query which is 10,000 documents evaluated in total. The generated subtopics and relevance labels of the documents were not available to any diversification systems and were stored in the judgment file for the evaluation of diversification methods.

We now discuss how to construct the concept hierarchy based on the structured information, i.e., relational databases. Since most queries in the query log are related to HP products, the structured information used in this paper is mainly about the products. However, the proposed methods can be applied on any other enterprise databases and

collections. In particular, we use the data from five tables containing the most important information of the products. The tables are *product type, product marketing category, product marketing subcategory, product series, and product name*. These tables are connected through foreign/primary key relationships, and these relationships can be used to construct a 5-level concept hierarchy, whose levels are *product type* containing 8 nodes, *product marketing category* with 54 nodes, *product marketing subcategory* with 134 nodes, *product series* with 983 nodes and *product name* with 3,238 nodes.

To diversify search results, we first use the default retrieval function to retrieve documents for each query from the document collection, and then extract the subtopics using different subtopic extraction methods. With the subtopics, the results are then diversified using the SQR method. Specifically, we use Indri to build index of the document collection. The preprocessing includes term stemming using Krovetz stemmer and stop words removal using the stop words provided by Indri. We use query likelihood generation retrieval function with Dirichlet smoothing (Zhai and Lafferty 2001) as the default retrieval function to retrieve documents and compute the components in Eq. (1). The parameter μ in Dirichlet is set to 2,500, which is the default value used in Indri.

We implement the following subtopic extraction methods:

- *PLSA*, which applies PLSA method to identify topics from retrieved documents, and then uses terms from topics as query subtopics as described in Sect. 3.1;
- *DB*, which extracts subtopics from the database using the method described in Sect. 3.2;
- *QuerySugg*, which uses top *M* suggested queries from Yahoo! search as query subtopics;
- *StruGuide*, the integration method described in Sect. 4.1;
- *UnstruGuide*, the method described in Sect. 4.2;
- *Combine*, the method described in Sect. 4.3

The diversification result of each set of subtopics is evaluated with the official measure used in the TREC diversity task Clarke et al. 2009a, 2010). The primary measure is a variant of intent-aware expected reciprocal rank (ERR-IA@20) (Chapelle et al. 2009), and other measures include $\alpha - nDCG@20$ (Clarke et al. 2008) and *NRBP* (Clarke et al. 2009b).

5.1.2 Effectiveness of subtopic integration methods

Table 1 reports the optimal performances of the diversification methods with the results of Wilcoxon test. We also report the results of fivefold cross-validation in Table 2, where all parameters are optimized at the same time. The results show that the subtopics extracted from both documents and databases are more effective than those from individual sources, and *Combine* is the the most effective method to combine the subtopics. Moreover, it is interesting to see that *QuerySugg* performs worse than *PLSA*, which is inconsistent with the previous findings on Web search (Santos et al. 2010c). This could be caused by the information gap between the Web query logs and the enterprise data collection. Note the performance improvement of *Combine* method over *QuerySugg* is also statistically significant.

It is clear that using query subtopics generated by the proposed integration methods can lead to better diversification performance than using those from PLSA. Through further analysis on subtopics, we find that better diversification performance is indeed related to the better quality of subtopics. Let us look at the results of query "HP PSC 1210".

Table 1 Optimal performance (enterprise), where $^{\blacklozenge}$, $^{\blacktriangle}$,‡ denote the methods are significantly better than *PLSA*, *DB* and *QuerySugg* at 0.05 level, respectively, $^{\diamond}$, $^{\Delta}$ and † denote the methods are significantly better than *PLSA*, *DB* and *QuerySugg* at 0.1 level, respectively

Methods	ERR-IA@20	α-nDCG@20	NRBP
PLSA	0.249	0.396	0.208
DB	0.239	0.391	0.202
QuerySugg	0.244	0.390	0.205
StruGuide	$0.262^{\diamond}$	$0.415^{\diamond}$	$0.224^{\diamond}$
UnstruGuide	$0.277^{\blacklozenge\Delta\dagger}$	$0.423^{\blacklozenge\Delta}$	$0.241^{\diamond\Delta\dagger}$
Combine	$\mathbf{0.292}^{\blacklozenge\blacktriangle\dagger}$	$\mathbf{0.443}^{\blacklozenge\blacktriangle\ddagger}$	$\mathbf{0.255}^{\blacklozenge\blacktriangle\ddagger}$

Bold values indicate the performances of the best method

Table 2 Cross-validation results (enterprise, ERR-IA@20)

Methods	Train		Test	
	Average	Deviation	Average	Deviation
PLSA	0.257	0.011	0.189	0.061
DB	0.240	0.009	0.231	0.051
QuerySugg	0.244	0.007	0.237	0.029
StruGuide	0.264	0.009	0.233	0.041
UnstruGuide	0.280	0.019	0.222	0.079
Combine	**0.293**	0.011	**0.269**	0.053

Bold values indicate the performances of the best method

According to the judgments, the query has five subtopics, i.e., "HP PSC 1210 all-in-one printer customer care", "HP PSC 1200 support and troubleshooting", "HP PSC 1200 software and driver downloads", "HP PSC 1200 manuals" and "HP PSC 1200 series product specifications". Table 3 shows the subtopics extracted by *Combine* and *PLSA*. We see that subtopics extracted by *Combine* method contain more relevant terms and cover more real subtopics of the query. For example, *service* and *contact* are related to the real subtopic "customer service", *report* and *solution* are related to "troubleshooting", *software* and *driver* are related to "driver downloads", *install* and *reinstall* are related to "manuals", and *psc 1216* and *cartridge* are related to "1200 series specification".

Table 3 Extracted Subtopics for query "*HP PSC 1210*"

Subtopic	*Combine*	*PLSA*
1	**print**, **report**, **contact**, visitor, **ink**	**explain**, 131, selection, produce, occurred
2	all, **service**, cp3520, **reinstsall**, 4050t	1320, **refurbished**, glitch, c8165c, c6409b
3	**software**, **laptop**, geekless, swood1122, 53ghz	cc335a, mercado, **unresolved**, eset, **capacity**
4	**solution**, image, **driver**, **cartridge**, **install**	**compatible**, especial, c9362he, cc567c, lib
5	fax, one, psc, **1216**, recommend	ch376b, greet, date, mart, 068
ERR-IA@20	0.721	0.144

Bold terms are related to real subtopics of the query

However, *PLSA* does not have terms related to "manuals" and "driver downloads". The diversification performance of the subtopics of *Combine* is better than that of subtopics of *PLSA* in this query.

Moreover, we also compare the subtopics extracted from the three integration method, and can make a few interesting observations. First, *Combine* can extract high-quality and non-overlapped subtopics that are effective in diversifying documents. Second, *UnstruGuide* tends to have more subtopics that are non-relevant to the query than *Combine* because there are many non-relevant documents in the original search results. Finally, *StruGuide* contain more noisy terms than those from *Combine*.

Finally, Tables 1 and 2 show that using the structured data alone, i.e., DB, is less effective than other methods. Our analysis suggests that this is caused by the vocabulary gap between the database and retrieved documents. There are two reasons for the existence of the vocabulary gap. (1) The retrieved documents contain relevant terms that are different from relevant terms in the database. For example, the subtopics extracted from the database contain *customer service* of the printer while web pages use terms *forum*, *solve problem* or *support* to describe the technique support. (2) The database we used is only a part the whole database of the company. Therefore, the structured data miss some relevant information which exists in the documents.

5.1.3 Parameter sensitivity

We examine how the parameters of the proposed subtopic extraction methods affect the diversification performance. The parameters include M, i.e., the number of subtopics used for diversification, γ in Eq. (2) and threshold described in Sect. 4.4 When examining the effect of one parameter, we tune the value of that parameter while keeping other parameters to be their optimal values. The optimal values of parameters are shown in Table 4.

γ is used in subtopic selection from structured data as shown in Eq. (2). When its value is small, it is more likely to select nodes at high level from the concept hierarchy as subtopics. Figure 6 shows the performance sensitivity with respect to its value. We can see that the performances change very little when γ is larger than 1.5. The optimal values for *StruGuide* and *Combine* are smaller than that for *UnstruGuide*, which indicates that *UnstruGuide* prefers to select nodes from a lower concept hierarchy. Recall that one difference between *UnstruGuide* and other two methods is that it selects nodes using retrieved documents instead of queries. Since the documents are much longer than the queries, they provide more detailed information and enable the matching with the nodes with more specific information, i.e, the ones at the lower level in the concept hierarchy.

M is the number of subtopics we used for diversification. *UnstruGuide* method does not have this parameter, because the number of subtopics used in that method is dynamically decided based on the matching between the relevant documents and the nodes. Figure 7

Table 4 Optimal values of parameters based on ERR-IA@20 (Enterprise)

Methods	M	γ	Threshold
PLSA	13	–	–
DB	9	0.2	–
QuerySugg	3	–	–
StruGuide	3	0.8	0.04
UnstruGuide	–	1.5	0.03
Combine	9	0.6	0.04

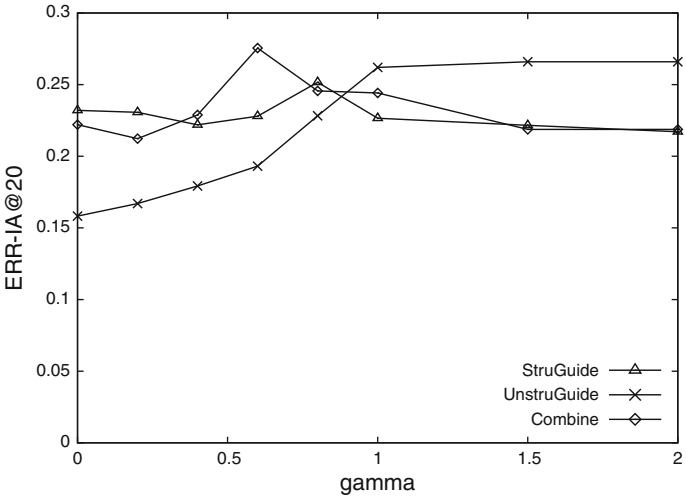

Fig. 6 Impact of γ (Eq. 2) on the diversification performances

shows the sensitivity plot. It is interesting to observe that *Combine* can often perform better than *StruGuide* when the number of subtopics is larger than 6 while worse when it is smaller.

The threshold described in Sect. 4.4 is used to control, for each query, whether we will use the information from structured data to extract subtopics. In each query, we use the subtopics extracted using proposed method to diversify results if the average relevance score of the subtopics to the query is larger than the value of the threshold. Otherwise, we use DOC subtopics that are only from unstructured data with PLSA and diversify results. Figure 8 shows the sensitivity plot. The systems use the integrated subtopics in more queries when the threshold is smaller while using DOC subtopics in more queries when the

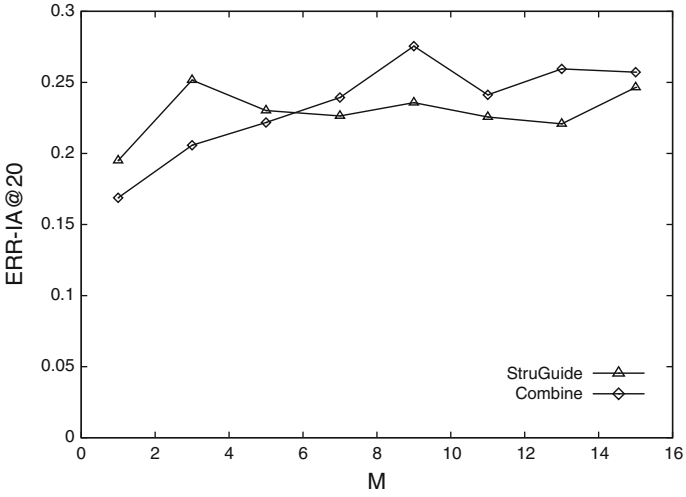

Fig. 7 Impact of M (i.e., number of subtopics) on the diversification performances

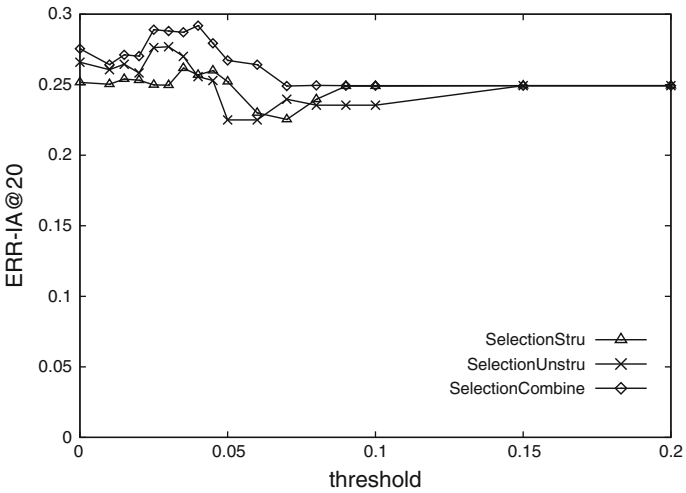

Fig. 8 mpact of the threshold (Sect. 4.4) on the diversification performances

threshold is larger. The optimal values are around 0.04. It is interesting to see that optimal performances when applying the threshold are both better than the integration methods, i.e., the leftmost points, and better than PLSA, i.e., the rightmost points. Therefore, it is important to predict the quality of the structured data in each query when extracting subtopics using the integrated information.

5.2 Effectiveness in web search

5.2.1 Experiment design

We also conduct the experiments in Web search domain. In particular, we leverage the efforts of TREC Web track and use the collections from the diversity task in TREC 2009 and 2010. Each collection has 50 queries. According to the judgment file, each query has 4.61 subtopics on average on TREC 2009 and 4.36 subtopics on TREC 2010. The relevance judgment is made with respect to the subtopics. The document collection is ClueWeb Category B data set, and contains 50 million English-language pages. TREC collection does not have structured data. We therefore use the ODP as the structured data with concept hierarchy.

5.2.2 Experimental results

We trained the parameters on TREC 2009 collection, and used the trained parameters to evaluate the testing performance on TREC10 collection. Table 5 shows both the training and testing performance and Table 6 shows the values of parameters trained on TREC 2009 collection.

The results show that the proposed *Combine* method outperforms all other methods, which indicates that the integrated subtopics are more effective than those extracted from individual sources. The performance improvements of *Combine* over all the three baselines are statistically significant on the testing collection. The testing performance on TREC 2010 can be ranked among top 4 runs in the diversity task.

Table 5 Performance trained on TREC 2009 and tested on TREC 2010, where $^{\blacklozenge,\blacktriangle,\ddagger}$ denote the methods are significantly better than *PLSA*, *DB* and *QuerySugg* at 0.05 level, respectively, $^{\diamond,\triangle}$ and † denote the methods are significantly better than *PLSA*, *DB* and *QuerySugg* at 0.1 level, respectively

Methods	Training on TREC 2009			Testing on TREC 2010		
	ERR-IA@20	α-nDCG@20	NRBP	ERR-IA@20	α-nDCG@20	NRBP
PLSA	0.154	0.253	0.129	0.171	0.262	0.137
DB	0.153	0.248	0.128	0.169	0.259	0.132
QuerySugg	0.167	0.258	0.142	0.185	0.272	0.152
StruGuide	**0.193**$^{\blacklozenge\triangle}$	**0.281**$^{\diamond}$	**0.177**$^{\blacklozenge}$	0.187	0.277	0.151
UnstruGuide	0.173	0.266	0.149	0.184	0.271	0.151
Combine	0.181$^{\diamond\blacktriangle\dagger}$	0.273$^{\triangle}$	0.161$^{\blacklozenge\blacktriangle\dagger}$	**0.208**$^{\blacklozenge\blacktriangle\ddagger}$	**0.305**$^{\blacklozenge\blacktriangle\ddagger}$	**0.172**$^{\blacklozenge\blacktriangle\dagger}$

Bold values indicate the performances of the best methods

Table 6 Optimal values of parameters based on ERR-IA@20 (TREC 2009)

Methods	M	γ	Threshold
PLSA	15	–	–
DB	5	0.4	–
QuerySugg	3	–	–
StruGuide	15	0.8	0.21
UnstruGuide	–	1	0.23
Combine	9	1	0.06

It would be interesting to compare the results on both domains, and revisit our discussions described in Sect. 4.4 Results show that *Combine* performs the best over most of the collections, which indicates that this method can indeed leverage the advantages of different data sources. Moreover, *UnstruGuide* performs better than *StruGuide* on the enterprise collection, while it performs worse on the Web collections. This is also consistent with our hypothesis. The reason is that *UnstruGuide* is affected by the quality of structured data more than *StruGuide*, and the structured data complements the data collection better in the enterprise domain than the Web domain.

We also compare the parameter values trained on TREC 2009 with the optimal parameter values on the enterprise collection. It is interesting to see that the values for *Combine* method are consistent on these two types of collections. In particular, the number of subtopics is set to 9, γ is around 0.8 and the threshold is around 0.04. On the contrary, the parameter values for the other two integration methods are more sensitive to the collections.

As described in Sect. 4.4, all the three methods dynamically decide whether to use the integrated subtopics or DOC subtopics. Table 7 shows the number of queries that use integrated subtopics for each method and each collection. It is clear that more than half of all the queries can benefit from the use of integrated subtopics.

6 Related work

We have briefly discussed the most closely related work above. We provide more details for those studies and discuss other related work in this section.

Table 7 Numbers of queries that used integrated subtopics instead of DOC subtopics. Note: the number of queries is 50 on each collection

Methods	Enterprise	TREC 2009	TREC 2010
UnstruGuide	30	35	25
StruGuide	31	36	30
Combine	20	49	46

Existing studies on diversification diversify search results either based on document redundancy (Carbonell and Goldstein 1998; Demidova et al. 2010; Zhai et al. 2003), or subtopic coverage (Agrawal et al. 2009; Carterette and Chandar 2009; Radlinski and Dumais 2006; Santos et al. (2010a, b, c). And the latter one has been shown to be more effective on TREC collections (Clarke et al. 2009a). Subtopic extraction is an important step in subtopicbased diversification, and various resources have been utilized to identify query subtopics.

The most commonly used resource is the document collection itself. Subtopics are extracted based on the clustering results of the pseudo relevant documents, i.e., top ranked documents in the retrieval result of the query (Balog et al. 2009a; Bi et al. 2009; Carterette and Chandar 2009; Dou et al. 2009; Li et al. 2009; Lubell-Doughtie and Hofmann 2011). One limitation of using the information from the unstructured documents is that the extracted subtopics contain lots of noisy terms which may hinder the diversification performance.

Taxonomies, such as ODP, have also been exploited for subtopic extraction. Hauff and Hiemstra used 98 categories in ODP and selected the categories that co-occurred with the query in the retrieved documents as the subtopics (Hauff and Hiemstra 2009). Other studies connected a query with top-level ODP categories through either query classification (Agrawal et al. 2009) or relevant Wikipedia pages (Santos et al. 2010b). One advantage of these taxonomies is that they contain high quality information without many noises. However, their effectiveness could be hindered by the vocabulary gap between the taxonomy and retrieved documents.

Query suggestions from Web search engines (Balog et al. 2009b; Li et al. 2009; Radlinski and Dumais 2006; Santos et al. 2010a) and the query log (Song et al. 2011) are also useful resources for subtopic extraction. Given a query, the studies used suggested queries provided by Web search engines or directly found related queries from the query log as the subtopics of the query. The advantage of this method is that the extracted query subtopics can represent different user search intents. However, its limitation is that the subtopics are independent of the document collection. When these subtopics are not good representation of relevant information in the document collection, they would not be effective to diversify search results. For example, query suggestions from Web search engines might not be effective to diversify enterprise search results.

Integrating subtopics from multiple sources has been recently studied (Dang et al. 2011; Dou et al. 2011; Radlinski et al. 2010). One strategy is to extract multiple subtopic sets from different sources and then combine them (Dou et al. 2011), and the other strategy is to directly extract a single set of subtopics from multiple sources (Dang et al. 2011; Radlinski et al. 2010; Zhang et al. 2011). These studies focused on Web search and used Web-related unstructured resources such as anchor texts, unigrams from Microsoft Web N-gram Services, user clicks, Web query log and related Web sites. It remains unclear how to extract and integrate subtopics from both unstructured and structured information. On the contrary, our paper focuses on using not only unstructured documents such as web pages but also structured information such as relational databases to extract query

subtopics. We also studied different methods to integrated subtopics extracted from these two types of information.

Our work is also related to result diversification over the relational databases (Chen and Li 2007; Demidova et al. 2010; Vee et al. 2008). Chen and Li returned a navigation tree based on the clusters of query log for users to select information that meets their needs of the query (Chen and Li 2007). This method relied on users to select the subtopics instead of automatically generating subtopics covering different user needs. Another study interpreted queries using the database schema and pre-defined query template (Demidova et al. 2010). They then iteratively selected the interpretation, i.e., subtopic, that is similar to the query and different from previously selected interpretations. They only used the content in the database. On the contrary, we use the concept hierarchy indicated by the relations among schemas together with the content to improve the effectiveness of subtopic extraction.

7 Conclusions and future work

This article presents the first study that looked into how to better leverage the integrated information from both the structured data and unstructured data to improve diversification performance. In particular, we focus on subtopic extraction step of search result diversification. We proposed the methods to integrate the structured and unstructured data to extract subtopics that are effective to diversify originally retrieved documents. We propose one method to effectively extract subtopics from structured data and three integration methods. The first integration method *StruGuide* uses subtopics extracted from the database as the prior of PLSA to guide subtopic extraction in documents. The second method *UnstruGuide* directly uses the documents to extract the most similar nodes on the concept hierarchy of the database. The third method *Combine* combines the subtopics that are separately extracted from database and documents. It first connects each subtopic of documents with the most similar subtopic of database. It then select terms of the subtopics of documents that are most similar to the connected subtopics of structured data. The experimental result shows that *Combine* is more effective and robust across different collections.

This paper focuses on diversifying documents to meet the different needs of users. However, the retrieval results in enterprise search may contain data from multiple repositories, e.g., database, email, RMS, etc. Therefore, it is necessary to re-rank them and contain data from different repositories in the top-ranked results. The data contained in one repository may be more important to one kind of queries and should be ranked higher while less important to other queries. Our proposed methods did not consider the importance of the repositories that the subtopics or data come from. In the future work, we will study how to combine subtopics and re-rank data of different repositories based on the importance of the repositories and relevance of these data. We will also explore new integration methods. For example, we can iteratively use subtopics from documents and subtopics from the database as the prior of each other. Finally, it would also be interesting to study how to construct reliable and re-usable evaluation collections that can directly evaluate the effectiveness of query subtopics, which is in the same line as INTENT task in NTCIR (Song et al. 2011; Sakai and Song 2012).

Acknowledgments This material is based upon work supported by the National Science Foundation under Grant Number IIS-1017026. We thank the journal reviewers for their useful comments.

References

Agrawal, R., Gollapudi, S., Halverson, A., & Leong, S. (2009). Diversifying search results. In *Proceedings of WSDM'09*.

Balog, K., Bron, M., He, J., Hofmann, K., Meij, E., de Rijke, M., et al. (2009a). The University of Amesterdam at TREC 2009. In *Proceedings of TREC'09*.

Balog, K., de Vries, A. P., Serdyukov, P., Thomas, P., & Westerveld, T. (2009b). Overview of the TREC 2009 entity track. In *Proceedings of TREC'09*.

Bi, W., Yu, X., Liu, Y., Guan, F., Peng, Z., Xu, H., et al. (2009). ICTNET at web track 2009 diversity task. In *Proceedings of TREC'09*.

Carbonell, J., & Goldstein, J. (1998). The use of MMR, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of SIGIR'98*.

Carterette, B., & Chandar, P. (2009). Probabilistic models of novel document rankings for faceted topic retrieval. In *Proceedings of CIKM'09*.

Chapelle, O., Metzzler, D., Zhang, Y., & Grinspan, P. (2009). Expected reciprocal rank for graded relevance. In *Proceedings of CIKM'09*.

Chen, Z., & Li, T. (2007). Addressing diverse user preferences in sql-query-result navigation. In *Proceedings of SIGMOD'07*.

Clarke, C. L. A., Craswell, N., & Soboroff, I. (2009a). Overview of the TREC 2009 web track. In *Proceedings of TREC'09*.

Clarke, C. L. A., Craswell, N., Soboroff, I., & Cormack, G. V. (2010). Overview of the TREC 2010 web track. In *Proceedings of TREC'10*.

Clarke, C. L. A., Koll, M., & Vechtomova, O. (2009b). An effectiveness measure for ambiguous and underspecified queries. In *Proceedings of ICTIR'09*.

Clarke, C. L. A., Kolla, M., Cormack, G. V., Vechtomova, O., Ashkann, A., Buttcher, S., et al. (2008). Novelty and diversity in information retrieval evaluation. In *Proceedings of SIGIR'08*.

Dang, V., Xue, X., & Croft, W. B. (2011). Inferring query aspects from reformulations using clustering. In *Proceedings of NTCIR-9*.

Demidova, E., Fankhauser, P., Zhou, X., & Nejdl, W. (2010). Divq: Diversification for keyword search over structured databases. In *Proceedings of SIGIR'10*.

Dou, Z., Chen, K., Song, R., Ma, Y., Shi, S., & Wen, J.-R. (2009). Microsoft research Asia at the web track of TREC 2009. In *Proceedings of TREC'09*.

Dou, Z., Hu, S., Chen, K., Song, R., & Wen, J. R. (2011). Multi-dimensional search result diversification. In *Proceedings of WSDM'11*.

Fang, H., & Zhai, C. (2006). Semantic term matching in axiomatic approaches to information retrieval. In *Proceedings of SIGIR'06*.

Hauff, C., & Hiemstra, D. (2009). University of Twente @ TREC 2009: Indexing half a billion web pages. In *Proceedings of TREC'09*.

Hawking, D. (2004). Challenges in enterprise search. In *Proceedings of ADC'04*.

He, J., Meij, E., & de Rijke, M. (2010). Result diversification based on query-specific cluster ranking. *Journal of the American Society for Information Science and Technology, 62*(3), 550–571.

Li, Z., Cheng, F., Xiang, Q., Miao, J., Xue, Y., Zhu, T., et al. (2009). THUIR at TREC 2009 web track: Finding relevant and diverse results for large scale web search. In *Proceedings of TREC'09*.

Lubell-Doughtie, P., & Hofmann, K. (2011). Improving result diversity using probabilistic latent semantic analysis. In *Proceedings of DIR'11*.

Mei, Q., Ling, X., Wondra, M., Su, H., & Zhai, C. (2007). Topic sentiment mixture: Modeling facets and opinions in weblogs. In *Proceedings of WWW'07*.

Radlinski, F., & Dumais, S. T. (2006). Improving personalized web search using result diversification. In *Proceedings of SIGIR'06*.

Radlinski, F., Szummer, M., & Craswell, N. (2010). Inferring query intent from reformulations and clicks. In *Proceedings of WWW'10*.

Sakai, T., & Song, R. (2012). Diversified search evaluation: Lessons from the NTCIR-9 INTENT task. *Information Retrieval*. doi:10.1007/s10791-012-9208-x.

Santos, R. L. T., Macdonald, C., & Ounis, I. (2010a). Exploiting query reformulations for web search result diversification. In *Proceedings of WWW'10*.

Santos, R. L. T., Macdonald, C., & Ounis, I. (2010b). Selectively diversifying web search results. In *Proceedings of CIKM'10*.

Santos, R. L. T., Peng, J., Macdonald, C., & Ounis, I. (2010c). Explicit search result diversification through sub-queries. In *Proceedings of ECIR'10*.

Song, R., Zhang, M., Sakai, T., Kato, M. P., Liu, Y., Sugimoto, M., et al. (2011). Overview of the ntcir-9 intent task. In *Proceedings of CIKM'11*.

van Rijsbergen, C. J. (1979) *Information retrieval*. Strand London: Butterworths.

Vee, E., Srivastava, U., Shanmugasundaram, J., Bhat, P., & Yahia, S. A. (2008). Efficient computation of diverse query results. In *Proceedings of ICDE'08*.

Zhai, C., Cohen, W., & Lafferty, J. (2003). Beyond independent relevance: Methods and evaluation metrics for subtopic retrieval. In *Proceedings of SIGIR'03*.

Zhai, C., & Lafferty, J. (2001). A study of smoothing methods for language models applied to ad hoc information retrieval. In *Proceedings of SIGIR'01*.

Zhang, S., Lu, K., & Wang, B. (2011). ICTIR subtopic mining system at NTCIR-9 INTENT task. In *Proceedings of NTCIR-9*.

Zheng, W., Fang, H., Yao, C.,& Wang, M. (2011a). Search result diversification for enterprise search. In *Proceedings of CIKM'11*.

Zheng, W., Wang, X., Fang, H., & Cheng, H. (2011b). An exploration of pattern-based subtopic modeling for search result diversification. In *Proceedings of JCDL'11*.

Zheng, W., Wang, X., Fang, H., & Cheng, H. (2012). Coverage-based search result diversification. *Information Retrieval, 15*(5), 433–457.