Subtopic Ranking Based on Block-Level Document Analysis

Tomohiro Manabe* and Keishi Tajima

Graduate School of Informatics, Kyoto University, Sakyo, Kyoto 606-8501 Japan
manabe@dl.kuis.kyoto-u.ac.jp,tajima@i.kyoto-u.ac.jp

Abstract. We propose methods for ranking subtopics of a keyword query. Subtopics are also keyword queries which specialize and/or disambiguate search intent behind their original query. Information on subtopics are useful for search systems to generate diversified search results. Search result diversification is important when there are multiple ways to interpret the submitted query. In search result diversification, it is important to rank subtopics by their intent probabilities that users need information on the subtopics. Our subtopic ranking methods use hierarchical structure in documents in the corpus. Hierarchical structure in documents consists of nested logical blocks with headings. A heading describes the topic of a part of a document, and a block is such a part of a document. All our methods are based on two assumptions related to the structure. First, hierarchical headings in a document represent hierarchical topics discussed in the document. Second, authors write more contents about subtopics with higher intent probabilities. Based on these assumptions, our methods score each subtopic based on the total size of the blocks whose hierarchical headings represent the subtopic. We develop our methods in the following way. We first propose four methods to score a subtopic on a document, four methods to integrate subtopic scores on multiple documents, and two methods to sort subtopics based on their scores. We then combined these methods, which results in 32 subtopic ranking methods in total. We evaluated these methods on the data set for the subtopic mining subtask of the NTCIR-10 INTENT-2 task. The results indicated that our methods generated rankings statistically significantly better than the query completion snapshots by major commercial search engines.

Keywords: Web Search, Search Result Diversification, Search Intent, Subtopic Mining, Hierarchical Heading Structure

1 Introduction

The Web is now one of the most important information resource, and the most standard way to obtain information from the Web is to submit a query consisting of keywords to Web search engines. Such keyword queries are sometimes ambiguous and/or referring to broad topics. For

* Current affiliation: Yahoo Japan Corporation, Chiyoda, Tokyo 102-0094 Japan

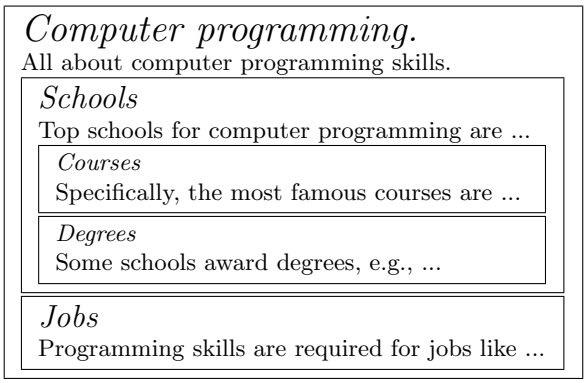


Fig. 1. Example web page with hierarchical heading structure. Each rectangle encloses block and each emphasized text is heading. Long texts are replaced by dots.

such queries, search result diversification techniques have been developed [7, 9, 24]. These techniques generate a page ranking including various topics so that it satisfies all information needs behind the given query. Subtopic mining is one of the most promising approaches to search result diversification [7]. Diversification methods based on subtopic mining first extract subtopic candidates of queries, then score and rank the subtopic candidates by their importance and distinctness, and finally returns a few pages each for the highly-ranked subtopic candidates. Because of the importance of subtopic mining, competitions for subtopic mining methods have been held as the subtopic mining subtasks of the NTCIR INTENT/IMine tasks [14, 23, 25, 36].

In this paper, we focus on important structure in documents which we consider is highly related to the problem of subtopic mining: *hierarchical heading structure*. Most documents contain hierarchical heading structure reflecting their topic structure. Hierarchical heading structure consists of nested logical blocks and each block includes its own heading. A heading represents the topic of its associated block and the hierarchical descendant blocks of the block. Because of this feature of heading, hierarchical headings in a document reflect topic structure in the document. For example, Figure 1 shows an example web page about "computer programming" (one of the NTCIR queries) containing hierarchical heading structure. In this figure, each rectangle encloses a block and each emphasized text is a heading. The hierarchical headings in this page reflect its topic structure. For example, its first level topic is computer programming, second level topics are computer programming schools and computer programming jobs, and the third level topics are courses and degrees of computer programming schools. Hierarchical heading structure of web pages are not obvious in general, but we have recently developed a method for extracting it [16].

In this paper, we propose methods to score hierarchical blocks in documents then rank subtopic candidates based on the block scores. To the best of our knowledge, this is the first paper which discusses the use of

detailed hierarchical heading structure in web pages for subtopic mining. Our basic ideas are that hierarchical headings in documents reflect hierarchical topic structure in the documents, and that more contents about a topic suggests more importance of the topic. Our methods score blocks based on the quantity of their contents, then approximate the importance of a subtopic candidate by the summation of its matching blocks' scores in a corpus. A subtopic candidate matches a block if the hierarchical headings of the block represent the candidate. To diversify resulting rankings, our methods adopt a subtopic with the best score one-by-one, and every time a subtopic is adopted, our methods re-score all remaining blocks after removing blocks matching subtopics which have been already adopted. By this approach, if some remaining subtopic candidates are already referred to by the blocks matching the already-adopted subtopics, or in other words, if some remaining subtopic candidates seems to be sub-subtopics of the already-adopted subtopics, the candidates lose their scores and resulting subtopic rankings get diversified.
The remainder of this paper is organized as follows. In the next section, we clarify our research targets. After that, we concisely survey related work. We then explain our methods in Section 4. In Section 5, we evaluate our methods on a publicly available NTCIR data set and compare the evaluation results with the baselines generated by major commercial web search engines. Lastly, Section 6 concludes this paper.

2 Definitions

In this section, we clarify the definitions of our research targets. They are namely subtopics of keyword queries and hierarchical heading structure in documents.

2.1 Definition of Subtopics

We focus on subtopics explicitly represented by subtopic strings defined in the NTCIR-10 INTENT-2 task [23] as quoted below.

> A subtopic string of a given query is a query that *specializes and/or disambiguates* the search intent of the original query. If a string returned in response to the query does neither, it is considered incorrect.

As defined above, each subtopic is associated to the original topic behind an original query. In INTENT-2 and in this paper, a *query* means a keyword query, which is an array of one or more words.
The overview paper of INTENT-2 lists some example subtopic strings [23]. If the original query is "harry potter", "harry potter philosophers stone movie" is a true subtopic string which specializes the original query. On the other hand, "harry potter hp" is not a subtopic string because hp is just the acronym of harry potter and the string neither specializes nor disambiguates the original query. If the original query is "office", "office workplace" is a subtopic string that disambiguates the original query considering the existence of office software, but "office office" is not a subtopic string. Note that true subtopic strings may not include

their original query. For example, “aliens vs predators” is a true subtopic string of the original query “avp” because avp can be an acronym of multiple terms.

2.2 Definition of Heading Structure

For ranking of subtopics, we use hierarchical heading structure in documents. We use our previous definition of the structure and its components given in [16], which is summarized below.

Heading A *heading* is a visually prominent segment of a document describing the topic of another segment.

Block A *block* is a coherent segment of a document which has its own heading describing its topic. As explained above, there is one-to-one correspondence between a heading and a block. We consider neither a block that consists only of its heading nor a block without its heading. This is because our research interest in this paper is the relationship between headings representing subtopics and blocks of various lengths. An entire document is also a block because it is a clearly specified segment and we can regard its title or URL as its heading.

Hierarchical Heading Structure A block may contain another block entirely, but two blocks never partially overlap. Therefore, all blocks in a document form a hierarchical structure whose root is the *root block* representing the entire document. We call the structure *hierarchical heading structure*.

3 Related Work

Generally, a term *topic* has two meanings in informatics [10]. One is an implicit topic represented by a (fuzzy) set of terms [11, 12], and the other is an explicit topic represented by a short text like a keyword query. Our research target is explicit topics. In particular, we focus on subtopics of topics which are behind keyword queries input by users. For mining such subtopics, we need four component technologies. They are namely subtopic candidate extraction, feature extraction from subtopic candidates, and subtopic ranking and diversification based on the features. We survey related work on these technologies in this order.

3.1 Subtopic Candidate Extraction

This step is not the topic of this paper. However, we briefly survey related work on this step for reference.
Query completion/suggestion by search engines generates many related queries of the original queries. This is a very popular resource of subtopic

candidate strings [15, 27, 28, 30, 33, 34, 37], and the snapshots of them for INTENT-2 [23] is publicly available. We also use them as baseline subtopic rankings for evaluation later. Google Insights and Google keywords generator are similar services [34]. Raw query logs of search engines [2, 15, 28, 30, 33] must also be useful.

Disambiguation pages in Wikipedia contain multiple subtopics of many ambiguous article titles of Wikipedia, and are very well-organized by hand. Therefore, they are also a very popular resource of subtopic candidate strings [15, 30, 33, 34, 37]. Redirect pages and tables of contents in Wikipedia must also be useful [33].

Of course, search result documents themselves can be a resource of subtopic candidate strings. Methods based on words frequently occurring [20, 29, 30, 35, 39, 40], words frequently co-occurring with query keywords [32], pseudo-relevance feedback [2], syntactic patterns [13], search result summaries [34] have been proposed.

Titles [20, 35], anchor texts of in-links [10, 34], and explicitly tagged top-level headings (H1 nodes) of HTML documents [34] all describe the topics of the entire documents. Therefore, they may be important as subtopic candidate strings. Their idea is similar to ours, but they do not use detailed hierarchical heading structure, i.e., low-level headings and their associated blocks. In addition, we use it for ranking candidate subtopic strings in this paper, not for extracting the candidates.

The QDMiner system extracts *query dimensions* each of which refers to one important aspect of the original query [8]. The system is based on list extraction from web pages. Their idea of query dimension is highly relevant to the idea of subtopic, and therefore some existing methods extract them as components of subtopic candidate strings [1, 28]. Some methods use lexical databases as well [2, 30].

3.2 Subtopic Feature Extraction

Similarly to most existing document ranking methods, many existing methods of subtopic feature extraction are based on term frequency (TF) and/or document frequency (DF) of subtopic strings or their component terms [5, 13, 32, 35, 39]. TF of a string means the number of its occurrences in a document, and DF of a string means the number of documents which contain it. The occurrences in some types of document metadata, e.g., document titles, anchor text of in-links, and top-level headings, are more important than other occurrences [34, 35].

Similarity between subtopic candidate strings and their search result documents, or between subtopic candidate strings and their original queries, is a popular feature [5, 15, 19, 39]. Document coverage of a subtopic candidate string is the weighted summation of the scores of documents that both the string and its original query retrieved [13].

Distinctness entropy of subtopic candidate strings measures the distinctness among the document sets that the strings retrieved [13, 38]. The SEM group at INTENT-2 used the co-occurrence of subtopic candidate strings in query logs and the edit distance between the strings and their original queries [28].

Query-independent features like readability of subtopic candidate strings are also useful [28, 32].

3.3 Subtopic Ranking

Subtopic ranking is essential for filtering out noises and for ranking subtopic strings by their importance. The simplest way is to sort subtopic candidate strings in order of linear combination of features. As in the area of document ranking, however, more sophisticated functions like TFIDF (TF over DF) and BM25 [22] are also used [13, 28, 30, 32].
Many methods assign different weights for different sources of subtopic candidate strings [15, 34]. For example, the THUIR group at the IMine task of NTCIR-11 assigned the weights of 0.75 for Google keywords generator, 0.15 for Google insights, and 0.05 for query completion/suggestion by commercial search engines [34].
Ullah and Aono proposed a method that represents each subtopic candidate string by its feature vector then score them by their cosine similarity with the mean vector [27].
It is notable that the THUSAM group at INTENT-2 adopted a variant of learning-to-rank methods that are state-of-the-arts methods for document ranking [15].

3.4 Subtopic Diversification

One important application of subtopic mining is search result diversification. Therefore, diversity of ranked subtopics is also important.
Subtopic diversification step is sometimes embedded into other steps. One promising way is clustering of subtopic candidate strings and extraction of the representative string of each cluster [13, 18, 29–31, 33–35, 37]. The cluster-level entropy maximization [13], affinity propagation [31, 33, 37], a variant of K-medoids [34], and K-means [35] algorithms are used. The THCIB group at NTCIR-10 clustered implicit topics by the affinity propagation algorithm, then assigned explicit topics to each cluster by Latent Dirichlet Allocation [31].
The Hierarchical InfoSimba-based Global K-means (HISGK-means) algorithm clusters search result snippets then labels each cluster [6, 18]. The InfoSimba is a similarity measure between snippets based on term co-occurrence, and HISGK-means recursively clusters snippets based on the measure and Global K-means. Each label is obtained as the centroid of a cluster.
Recently, some methods adopted word embedding models [15, 19]. In word embedding models, we can *subtract* subtopic candidate strings from their original query. Based on this idea, the HULTECH group at IMine recursively subtracted subtopic candidate strings from their original query then compared the difference and the remaining subtopic candidate strings every time they adopt the subtopic candidate string with the best score [19]. Their idea is similar to ours, except we recursively subtract blocks, not vectors representing words, from a web corpus.
The maximal marginal relevance (MMR) framework also concatenate items into rankings one-by-one [3]. In each iteration, the framework selects the item with the best balance of the score and dissimilarity to the already ranked items. Of course, the framework is useful for diversifying subtopic rankings [27].

As explained above, no existing method scores or diversifies subtopic candidate strings based on detailed logical hierarchical structure in documents, e.g. hierarchical heading structure, which our methods use.

4 Subtopic Ranking Based on Hierarchical Heading Structure

In this section, we propose scoring and ranking methods for subtopic candidate strings. Our proposed methods are based on matching between the strings and hierarchical heading structure in documents in a corpus. We regard that a subtopic candidate string *matches* a block if and only if all the words in the string appear either in the heading of the block or in the headings of its ancestor blocks (after basic pre-processing, i.e., tokenization, stop word filtering, and stemming). For example, a subtopic string "computer programming degrees" matches the "degrees" block in Figure 1 because the top-level heading of the block contains "computer" and "programming" and the own heading of the block contains the remaining word "degrees". If a subtopic string matches a block, the block must refer to the subtopic according to the definition of hierarchical heading structure. Because of this definition of matching, if a subtopic string matches a block, the string must also match the hierarchical descendant blocks of the block. However, we do not consider such matching of hierarchical descendants of already matched blocks. Instead, we score each block considering its hierarchical descendant blocks.
Formally, the score of a pair of a subtopic string s and a document d is:

$$\text{docScore}(s, d) = \sum_{b \text{ in } d} \text{match}(s, b)\text{blockScore}(b) \tag{1}$$

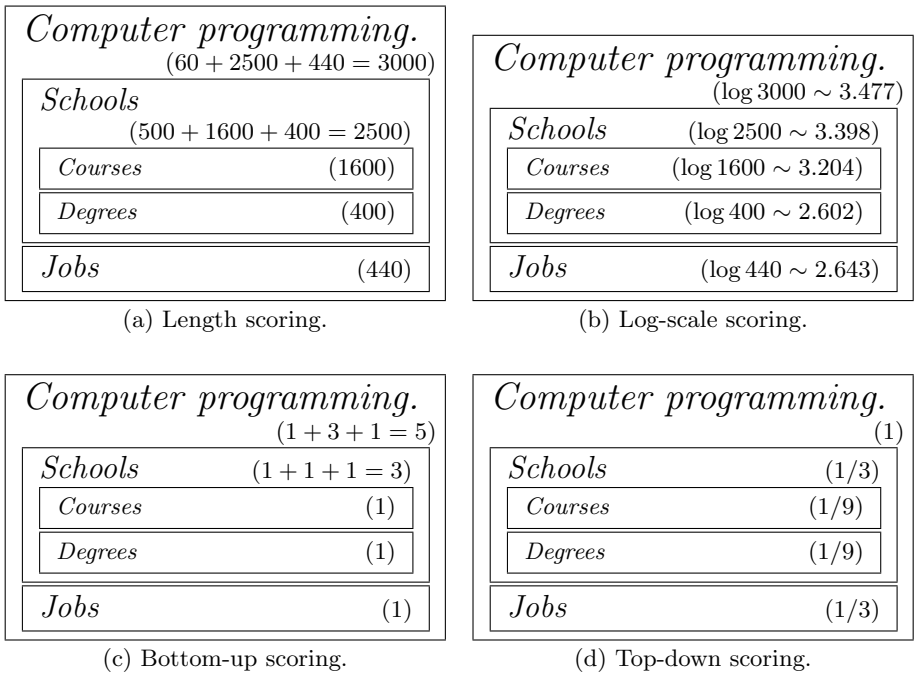
where b is each block in d, $\text{match}(s, b)$ is 1 if and only if s matches b and does not match any ancestor block of b and is 0 otherwise. $\text{blockScore}(b)$ is the score of b.
Hereafter in this section, we first discuss the definition of $\text{blockScore}(b)$, then discuss integration of subtopic scores on multiple documents, and finally discuss ranking of multiple subtopics into a diversified ranking.

4.1 Block Scoring

First, we propose four definitions of $\text{blockScore}(b)$.

Scoring by Content Length Basically, the more description about a subtopic a document contains, the more important the subtopic is for the document author. Furthermore, the importance of the subtopic for readers, and for search engine users, is also reflected by the length of the content because generally speaking authors write documents for readers. Based on this idea, we can score blocks by the lengths of their contents. The score of a block b is:

$$\text{blockScore}(b) = \text{length}(b) \tag{2}$$



(a) Length scoring. (b) Log-scale scoring.

(c) Bottom-up scoring. (d) Top-down scoring.

Fig. 2. Comparison of scoring results by four scoring methods of page in Figure 1. Scores of blocks are in parentheses. Non-heading components of blocks are omitted.

where length(b) is the length of b. We call this *length* scoring. For example, if we score the blocks in Figure 1 by this, we obtain the result shown in Figure 2a. In Figure 2, the scores of the blocks are in parentheses and non-heading components of the blocks are omitted.

Scoring by Log-Scaled Content Length As the relevance between a document and a query keyword is assumed to be not direct proportional to the number of the query keyword occurrences in the document [22], the importance of a topic may also be not direct proportional to the content length of the block referring to the topic. Based on this idea, we propose another scoring function with logarithmic scaling:

$$\text{blockScore}(b) = \log(\text{length}(b) + 1)\ . \tag{3}$$

We call this *log-scale* scoring. An example result of log-scale scoring is shown in Figure 2b.

Bottom-Up Scoring In practice, the importance of some topics are not reflected by the content length of their matching blocks. For example, telephone number may be an important subtopic of a place, but blocks under the heading "telephone number" should contain relatively

less contents, i.e., only the exact telephone number of the place, than blocks under other headings. Logarithmic scaling in the previous section reduces the effect of content length, but we also consider a scoring function that completely ignores content lengths. If we assume even importance for all blocks excluding their child blocks, the score of a block b is formulated as below:

$$\text{blockScore}(b) = 1 + \sum_{c \in b} \text{blockScore}(c) \tag{4}$$

where c is each child block of b. We call this *bottom-up* scoring. An example result of bottom-up scoring is shown in Figure 2c.

Top-Down Scoring On the other hand, we can assume even importance for all child blocks of a block. This assumption means that child blocks of a block are used to segment its topic into multiple subtopics of even importance. Because an original block may include meaningful contents besides its child blocks, we also assign the same importance to the contents. Formally, the score of a block b is:

$$\text{blockScore}(b) = \begin{cases} \dfrac{\text{blockScore}(p)}{1 + |p|} & \text{if } b \text{ has its parent block } p \\ 1 & \text{otherwise} \end{cases} \tag{5}$$

where $|p|$ is the number of the child blocks of p. We call this *top-down* scoring. An example result of top-down scoring is shown in Figure 2d.

4.2 Score Integration for Multiple Pages

Next, we explain four ways to integrate the scores of a subtopic candidate string s on each document d, $\text{docScore}(s, d)$, into $\text{score}(s, D)$, which is the score of the string on an entire document collection, or a corpus, D.

Integration by Simple Summation The simplest way to integrate the scores on multiple documents is to sum them up. Such simple summation means that the importance of a subtopic string is reflected by the length of contents (if we adopt length scoring), the number of blocks (if we adopt bottom-up scoring), and so on that refer to the subtopic in the corpus. Formally, the score of a subtopic string s on a corpus D is:

$$\text{score}(s, D) = \sum_{d \in D} \text{docScore}(s, d) \ . \tag{6}$$

We call this method *summation* integration.

Page-Based Integration In summation integration, documents of more length (if we adopt length or log-scale scoring) or including more blocks (if we adopt bottom-up scoring) have more chance to contribute

to $\mathrm{score}(s, D)$. However, if we assume that each document is equally important, the scaling of $\mathrm{docScore}(s, d)$ defined below may be useful:

$$\mathrm{score}(s, D) = \sum_{d \in D} \frac{\mathrm{docScore}(s, d)}{\mathrm{blockScore}(\mathrm{root}(d))} \tag{7}$$

where $\mathrm{root}(d)$ is the root block of d, i.e., the block representing the entire document d. We call this method *page-based* integration.
Because we score each block considering its hierarchical descendant blocks, $\mathrm{blockScore}(\mathrm{root}(d))$ takes its maximum value among all the blocks in d. This division by $\mathrm{blockScore}(\mathrm{root}(d))$ scales $\mathrm{docScore}(s, d)$ to $[0, 1]$ if we adopt length, top-down, or bottom-up scoring methods. Note that $\mathrm{docScore}(s, d)$ may exceed 1 if we adopt log-scale scoring and multiple blocks in d matches s.
Note that there is no difference between summation integration and page-based integration if we adopt top-down scoring because $\mathrm{blockScore}(b)$ in top-down scoring is already scaled to $[0, 1]$.

Domain-Based Integration Authors may split the contents about a topic into multiple documents in a *domain*, e.g., a set of web pages whose URLs include the same domain name, instead of splitting it into multiple blocks in a single document.
Considering such cases, domain-based scaling may be more effective than page-based scaling. To formulate such scaling, we introduce Δ, a set of domains which appear in the corpus. Each domain $\delta \in \Delta$ is a subset of the corpus D, and $\bigcup_{\delta \in \Delta} \delta = D$. A new integration function is:

$$\mathrm{score}(s, D) = \sum_{\delta \in \Delta} \frac{\sum_{d \in \delta} \mathrm{docScore}(s, d)}{\sum_{d \in \delta} \mathrm{blockScore}(\mathrm{root}(d))} \ . \tag{8}$$

We call this method *domain-based* integration.

Combination Integration If we apply both page-based and domain-based scalings, the new integration function is:

$$\mathrm{score}(s, D) = \sum_{\delta \in \Delta} \frac{1}{|\delta|} \sum_{d \in \delta} \frac{\mathrm{docScore}(s, d)}{\mathrm{blockScore}(\mathrm{root}(d))} \ . \tag{9}$$

We call this *combination* integration.

4.3 Diversifying Subtopic Ranking

Next, we explain two ways to rank multiple subtopics of a query with varied $\mathrm{score}(s, D)$ into a ranking for the query.

Uniform Ranking To rank multiple subtopic strings into a ranking, we can score each of them once, then simply sort the strings by descending order of their scores. We call this *uniform* ranking method.

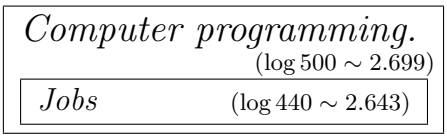


Fig. 3. Example re-scoring result of page in Figure 1 by log-scale scoring after we rank first subtopic string "computer programming schools".

Diversified Ranking However, because search result diversification is one of the most important applications of subtopic ranking, diversity of subtopic ranking is also important. Therefore, we also propose a diversification method for subtopic ranking. Our idea for the diversification is that if a block matches a subtopic candidate string which is already ranked into the ranking, the topic of the block is already referred to by the ranked subtopic string, and therefore, even if the block matches some other remaining subtopic candidate strings, the block should not contribute to the score of the candidate strings.

Based on this idea, we propose a *diversified* ranking method for subtopic strings based on hierarchical heading structure. In this method, first we score each subtopic candidate string on a corpus then put only the string with the best score into the resulting ranking. Second, we remove all the blocks matching the string from the corpus. Third, we again score the remaining subtopic candidate strings on the remaining blocks then put the string with the best score into the resulting ranking. The second and third steps are repeated until all the subtopic candidate strings are ranked or enough number of subtopics are ranked.

For example, suppose we have three subtopic strings, "computer programming school", "computer programming course", and "computer programming jobs". If we rank the strings by uniform ranking method and the log-scale scores of the blocks on the document in Figure 2b, the ranks of the strings are in the above order because the strings respectively match the "Schools" (score: 3.398), "Courses" (score: 3.204), and "Jobs" (score: 2.643) blocks. On the other hand, if we rank the strings by diversified ranking method, "computer programming jobs" occupies the second rank. This is because after "computer programming school" is ranked first, its matching block "School" including its descendant blocks is removed from the re-calculation of the scores. Then the score of "computer programming course" in this page becomes 0 because the "Courses" block referring to the subtopic candidate has already removed from this page.

5 Evaluation

In this section, we evaluate and compare baseline rankings and rankings generated with our proposed methods.

We proposed four block scoring methods, four score integration methods, and two subtopic ranking methods. We can arbitrary combine these methods. However, there is no difference between summation and page-based integration and also between domain-based and combination inte-

gration when we use top-down scoring as discussed in Section 4.2. Therefore, we compare 28 proposed methods in total.

5.1 Evaluation Methodology

Because we do not discuss extraction of subtopic candidate strings, we evaluate our proposed methods by re-ranking baseline subtopic rankings. We use the official data set, including the baselines, and the evaluation measures of the subtopic mining subtask of the NTCIR-10 INTENT-2 task [23]. This is because the dataset of the latest NTCIR-12 IMine-2 task [36] is not available yet, and because first-level and second-level subtopics are distinguished in the second-latest NTCIR-11 IMine task [14] while our proposed methods do not distinguish them. All components of the NTCIR-10 data set is publicly available and most of them are on the web site of NII[1].

In the subtopic mining subtask, participants are required to return ranked list of top-10 subtopic strings for each query. Subtopic strings are expected to be sorted in descending order of their *intent probability*, i.e. the probability that search engine users submitting the given query need information on the subtopics. Multiple subtopic strings may refer to the same subtopic, but a string refers to one subtopic at most.

Official evaluation measures of the subtask are intent recall (I-rec), D-nDCG, and D♯-nDCG.

The definition of the I-rec measure is:

$$\text{Irec@10} = \frac{|I'|}{|I|} \tag{10}$$

where I is a set of known subtopics of the original query, and I' is a set of subtopics represented by any of the maximum 10 strings in a ranking to be evaluated. This measure reflects recall and diversity of subtopics in rankings.

The definition of the D-nDCG measure is:

$$\text{DnDCG@10} = \frac{\text{DDCG@10}}{\text{ideal DDCG@10}} \tag{11}$$

$$\text{where DDCG@10} = \sum_{r=1}^{10} \frac{\sum_i Pr(i|q) g_i(r)}{\log(r+1)} \tag{12}$$

where r is a rank, $Pr(i|q)$ is the intent probability of a known subtopic i behind the original query q, and $g_i(r)$ is 1 iff the string at the rank r refers to the subtopic i, and 0 otherwise. The D-nDCG measure reflects the precision and accuracy of subtopics in rankings.

The integrated measure D♯-nDCG is the weighted summation of I-rec and D-nDCG.

$$\text{D♯nDCG@10} = \gamma \text{Irec@10} + (1-\gamma)\text{DnDCG@10} \tag{13}$$

where γ is the weight of I-rec which is fixed to 0.5 in this paper and the subtask. In other words, D♯-nDCG is arithmetic mean of I-rec and D-nDCG in this paper and the subtask.

[1] `http://www.nii.ac.jp/dsc/idr/en/ntcir/ntcir.html`

An official evaluation tool is available online[2].

5.2 Data Set

The details of the data set is as follows.

Queries We used 50 keyword queries in the NTCIR data set which are also used in the web track of the well-known Text Retrieval Conference (TREC) 2012 [4].

Document Sets We used the documents on baseline document rankings generated by default scoring of Indri search engine (including query expansion based on pseudo-relevance feedback) [26] and Waterloo spam filter. The baseline rankings are prepared for the TREC 2014 web track and contains rankings for the queries prepared for the TREC 2012–2014 web tracks. Each ranking consists of 131–837 web pages for a query extracted from the ClueWeb09B document collection, and we use them as the corpus for re-ranking the baseline subtopics of the query.
The baseline rankings are available online[3].
The ClueWeb09B document collection is one of the most well-known snapshots of the web, contains 50 million web pages, and is crawled by the Lemur Project in 2009.
The document collection is also available at distribution cost[4].

Baseline Subtopic Rankings The NTCIR data set includes snapshots of query completion/suggestion results by major commercial search engines. We used the query completion results by Google and Yahoo because they respectively achieved the best I-rec and D-nDCG scores among the baselines [23]. Because the both results contain only 10 strings at most for each query, re-ranking of them do not affect I-rec scores. Therefore, we also used *our merged baseline result* which is generated by merging all of the four baseline query completion/suggestion results and sorting them in "dictionary sort" [23]. Because the meaning of dictionary sort is ambiguous, we could not reproduce their evaluation result. We merged the results, decapitalized the strings, removed duplicated strings, and sorted the remaining strings in byte order in UTF-8 to generate our merged baseline result.

Known Intents and Intent Probabilities The actual known subtopics, subtopic strings referring to them, and their intent probabilities are manually prepared for the subtask [23]. Note that all the actual subtopic strings in the baseline subtopic rankings must be in this data according to their annotation process [23].

[2] `http://research.nii.ac.jp/ntcir/tools/ntcireval-en.html`
[3] `https://github.com/trec-web/trec-web-2014`
[4] `http://www.lemurproject.org/clueweb09/`

5.3 Implementation Details

In this section, we explain the details of our implementation required to evaluate our methods.

Heading Structure Extraction To extract hierarchical heading structure in web pages, we use our previously proposed heading-based page segmentation (HEPS) method [16]. It extracts each heading and block in pages as an array of adjoining sibling DOM nodes. For evaluation, we used the reference implementation 1.0.0 of HEPS[5].

Text Contents of Headings and Blocks We used the URL and the title as the heading of each web page. As the text contents of the other headings, blocks, and entire pages, we use their corresponding *raw strings* that we previously defined [16]. Intuitively, the raw string of a component is the string of the DOM text nodes in the component. Before generating raw strings, each DOM IMG (image) nodes are replaced by its alternate text and URL, i.e., alt and src HTML attribute values, to treat the IMG nodes as text nodes.

Content Length For length and log-scaled scoring, we used the number of UTF-8 characters in their raw strings as their length. Note that the documents in the ClueWeb09 collection are encoded in UTF-8.

Domain For domain-based and combination integration, we distinguished the domains of web pages by the domain names in their URLs.

Matching between Subtopic Strings and Headings Before matching subtopic candidate strings and hierarchical headings, we applied basic preprocessing for text retrieval, e.g., tokenization, stop word filtering, and stemming, to both types of strings. All URLs were tokenized by splitting by any non-word characters, and the other strings were tokenized by Stanford CoreNLP toolkit [17]. All tokens were decapitalized, filtered out if they are 33 default stop words of the Lucene library[6], and then stemmed by the Porter stemmer [21].

Subtopic Candidate Strings After preprocessing, duplicated subtopic candidate strings and subtopic candidate strings same as their original queries were removed.

Tie Breaking If we have multiple subtopic candidates of the same score in our unified ranking method or in any iteration of our diversified ranking method, we sorted them in the same order as the baseline subtopic ranking.

Table 1. Comparison with query completion result by Google. Our methods are listed in descending order of their D-nDCG scores. Best score is in bold font. For all methods and baseline, I-rec score is .3841.

Scoring	Integration	Ranking	D-nDCG
Log-Scale	Domain-Based	Uniformed	**.4502**
Log-Scale	Combination	Uniformed	.4501
Log-Scale	Domain-Based	Diversified	.4487
Log-Scale	Combination	Diversified	.4485
Bottom-Up	Page-Based	Diversified	.4479
Bottom-Up	Page-Based	Uniformed	.4474
Length	Combination	Uniformed	.4474
Log-Scale	Page-Based	Uniformed	.4474
Log-Scale	Summation	Diversified	.4470
Log-Scale	Page-Based	Diversified	.4470
Top-Down	Domain-Based	Uniformed	.4468
Top-Down	Combination	Uniformed	.4468
Log-Scale	Summation	Uniformed	.4467
Bottom-Up	Domain-Based	Uniformed	.4466
Top-Down	Summation	Diversified	.4460
Top-Down	Page-Based	Diversified	.4460
Length	Combination	Diversified	.4458
Length	Domain-Based	Uniformed	.4457
Bottom-Up	Combination	Uniformed	.4454
Length	Page-Based	Diversified	.4453
Top-Down	Page-Based	Uniformed	.4451
Top-Down	Summation	Uniformed	.4451
Top-Down	Domain-Based	Diversified	.4446
Top-Down	Combination	Diversified	.4446
Bottom-Up	Domain-Based	Diversified	.4446
Bottom-Up	Combination	Diversified	.4444
Length	Page-Based	Uniformed	.4442
Length	Domain-Based	Diversified	.4432
Length	Summation	Diversified	.4418
Length	Summation	Uniformed	.4416
Bottom-Up	Summation	Diversified	.4409
Bottom-Up	Summation	Uniformed	.4397
Query completion result of Google			.3735

Table 2. Comparison with query completion result by Yahoo. Our methods are listed in descending order of their D-nDCG scores. Best score is in bold font. For all methods and baseline, I-rec score is .3815.

Scoring	Integration	Ranking	D-nDCG
Log-Scale	Page-Based	Diversified	**.4617**
Bottom-Up	Domain-Based	Diversified	.4609
Log-Scale	Page-Based	Uniformed	.4608
Log-Scale	Summation	Diversified	.4601
Length	Domain-Based	Diversified	.4587
Bottom-Up	Domain-Based	Uniformed	.4585
Log-Scale	Summation	Uniformed	.4584
Top-Down	Domain-Based	Diversified	.4584
Top-Down	Combination	Diversified	.4584
Length	Combination	Diversified	.4583
Bottom-Up	Combination	Diversified	.4577
Top-Down	Domain-Based	Uniformed	.4569
Top-Down	Combination	Uniformed	.4569
Bottom-Up	Summation	Diversified	.4568
Length	Combination	Uniformed	.4566
Bottom-Up	Page-Based	Diversified	.4565
Bottom-Up	Combination	Uniformed	.4564
Top-Down	Summation	Diversified	.4562
Top-Down	Page-Based	Diversified	.4562
Length	Domain-Based	Uniformed	.4560
Log-Scale	Domain-Based	Diversified	.4557
Bottom-Up	Page-Based	Uniformed	.4557
Log-Scale	Combination	Diversified	.4551
Length	Summation	Diversified	.4549
Length	Page-Based	Diversified	.4549
Top-Down	Page-Based	Uniformed	.4548
Top-Down	Summation	Uniformed	.4548
Bottom-Up	Summation	Uniformed	.4541
Log-Scale	Domain-Based	Uniformed	.4537
Log-Scale	Combination	Uniformed	.4536
Length	Page-Based	Uniformed	.4528
Length	Summation	Uniformed	.4521
Query completion result of Yahoo			.3829

Table 3. Comparison with our merged baseline result. Our methods are listed in descending order of their D♯-nDCG scores. Best scores are in bold font.

Scoring	Integration	Ranking	I-rec	D-nDCG	D♯-nDCG
Log-Scale	Summation	Uniformed	**.4009**	**.3997**	**.4003**
Log-Scale	Page-Based	Uniformed	.3986	.3981	.3984
Length	Summation	Uniformed	.3974	.3945	.3959
Log-Scale	Combination	Uniformed	.3956	.3921	.3939
Log-Scale	Domain-Based	Uniformed	.3956	.3913	.3934
Length	Page-Based	Uniformed	.3974	.3882	.3928
Bottom-Up	Page-Based	Uniformed	.3918	.3930	.3924
Length	Combination	Uniformed	.3900	.3948	.3924
Top-Down	Combination	Uniformed	.3895	.3947	.3921
Top-Down	Domain-Based	Uniformed	.3895	.3947	.3921
Bottom-Up	Combination	Uniformed	.3880	.3944	.3912
Length	Domain-Based	Uniformed	.3855	.3930	.3893
Top-Down	Summation	Uniformed	.3872	.3906	.3889
Top-Down	Page-Based	Uniformed	.3872	.3906	.3889
Bottom-Up	Domain-Based	Uniformed	.3827	.3937	.3882
Top-Down	Combination	Diversified	.3869	.3710	.3790
Top-Down	Domain-Based	Diversified	.3869	.3710	.3790
Bottom-Up	Summation	Uniformed	.3726	.3824	.3775
Length	Summation	Diversified	.3855	.3682	.3768
Log-Scale	Page-Based	Diversified	.3840	.3695	.3768
Top-Down	Summation	Diversified	.3847	.3686	.3767
Top-Down	Page-Based	Diversified	.3847	.3686	.3767
Length	Combination	Diversified	.3836	.3693	.3764
Bottom-Up	Page-Based	Diversified	.3830	.3694	.3762
Bottom-Up	Combination	Diversified	.3813	.3707	.3760
Log-Scale	Summation	Diversified	.3812	.3694	.3753
Length	Page-Based	Diversified	.3852	.3639	.3746
Length	Domain-Based	Diversified	.3812	.3663	.3737
Log-Scale	Domain-Based	Diversified	.3813	.3659	.3736
Bottom-Up	Domain-Based	Diversified	.3780	.3681	.3731
Log-Scale	Combination	Diversified	.3813	.3640	.3727
Bottom-Up	Summation	Diversified	.3757	.3652	.3704
Our merged baseline result			.3310	.3066	.3188

5.4 Evaluation Results

Table 1, 2, and 3 show evaluation results. Table 1 shows the D-nDCG scores achieved by each method when they re-rank the query completion result by Google, and Table 2 shows the D-nDCG scores achieved by each method when they re-rank the query completion result by Yahoo. In Table 1 and 2, all our methods are listed in descending order of their D-nDCG scores. Table 3 shows the scores achieved by each method when they re-rank our merged baseline result. In Table 3, all our methods are listed in descending order of their D♯-nDCG scores.

5.5 Discussion

In all the comparisons, all our proposed methods consistently achieved scores better than the baseline scores on all the measures. This fact strongly supports the effectiveness of considering hierarchical headings and lengths of blocks for subtopic ranking. This consistency is due to a considerable number of subtopic candidate strings which were assigned score 0, and this effectiveness is due to such strings which are actually not subtopics or not important subtopics. For example, let us focus on the log-scale/page-based/diversified method which achieved the best D♯-nDCG score ($(0.3815 + 0.4617)/2 = \mathbf{0.4216}$) throughout this paper by re-ranking the query completion result by Yahoo. With this combination, 178 among 448 (39.7%) subtopic candidate strings were assigned score 0. In other words, no block in our corpus matched with these strings. Regardless of the choice of block scoring and score integration methods, these strings should be assigned score 0. Note that this fact does not indicate a flaw of our methods because they achieved the scores better than the baselines, and that larger corpus must support our methods to rank the zero-scored strings correctly.

Next, let us continue focusing on the log-scale/page-based/diversified method. The method also achieved its D-nDCG score (0.4470) better than the score of the query completion result by Google (0.3735) and its I-rec, D-nDCG, and D♯-nDCG scores (0.3840, 0.3695, and 0.3768, respectively) better than the scores of our merged result (0.3310, 0.3066, and 0.3188, respectively). Moreover, according to Student's paired t-test (where each pair consists of the scores of the baseline and our proposed method for a query), all the D-nDCG and D♯-nDCG scores were statistically significantly different from the baseline scores ($p < 0.05$). This fact supports the effectiveness of this combination of our proposed methods. Only the I-rec score was not statistically significant ($p = 0.0656$). Hereafter in this paper, we discuss statistical significance based on the same test procedure.

Comparison of Block Scoring Methods Log-scale scoring achieved the best scores in all the three comparisons. This fact may suggest that

[5] `https://github.com/tmanabe/HEPS`
[6] `http://lucene.apache.org/`

the importance of a topic is reflected by the content length of the block referring to the topic, but the importance is not direct proportional to the length. Moreover, 11 among the 15 best results shown in Table 1, 2, 3 are using log-scale scoring. This fact may suggest the robustness of log-scale scoring. However, the advantage of log-scale scoring over the others was small. For example, the D-nDCG score of the re-ranked Yahoo result by the *log-scale*/page-based/diversified method was not statistically significantly different from the scores of the *bottom-up*/page-based/diversified ($p = 0.1481$), *top-down*/page-based/diversified ($p = 0.1204$), and *length*/page-based/diversified ($p = 0.0972$) methods. To prove the advantage of log-scale scoring, we need more experiments on larger corpora.

Comparison of Score Integration Methods Score integration methods had only small impact. In the comparison with the Google result (Table 1), the log-scale/*domain-based*/uniform method achieved the best D-nDCG score, but its difference from the second-best score by log-scale/*combination*/uniform method was quite small (0.0001). In the comparison with the Yahoo result (Table 2), the log-scale/*page-based*/diversified method achieved the best D-nDCG score, but its difference from the score by the log-scale/*summation*/diversified method was also small (0.0016). In the comparison with our merged result (Table 3), the differences between the best log-scale/*summation*/uniform method and the second-best log-scale/*page-based*/uniform method were also small (I-rec@10: 0.0023, D-nDCG@10: 0.0016, D♯-nDCG: 0.0019). All these five differences were not statistically significant.
In this experiment, there was no substantial difference between our score integration methods.

Effect of Diversified Ranking Method Because I-rec can measure diversity of rankings, we focus on the I-rec score comparison with our merged result (Table 3). Unfortunately, no diversified method achieved its I-rec score better than 0.3869 while multiple uniformed methods achieved their I-rec scores better than 0.39.
In detail, the top-down/combination/*diversified* method achieved the best I-rec score (0.3869) among the methods with diversified ranking while the log-scale/summation/*uniformed* method achieved the best I-rec score (0.4009) among all the methods. However, the I-rec score difference between the methods was not statistically significant ($p = 0.2759$). The I-rec score difference between the best log-scale/summation/*uniformed* method and the log-scale/summation/*diversified* method was also not statistically significant ($p = 0.1028$).
The facts show that our proposed ranking diversification method did neither improve nor worsen resulting rankings.

6 Conclusion and Future Work

We proposed subtopic ranking methods based on the ideas that hierarchical headings in a document reflect the topic structure in the document

and that the length of contents referring to a topic reflects the importance of the topic. Based on these ideas, all our methods score subtopic candidate strings based on the lengths of the blocks whose hierarchical headings match the strings. Our methods consist of three steps: block scoring, integration of block scores, and ranking of subtopic candidate strings based on the integrated score of their matching blocks. We proposed four methods to score blocks, four methods to integrate block scores, and two methods to rank strings.

We evaluated our total 32 methods by using the publicly available NTCIR data set. The results indicated (1) our methods statistically significantly improved the baseline rankings by commercial search engines, (2) our corpus was not large enough for our methods to score less important subtopic strings correctly, (3) log-scale scoring seems effective and robust, (4) there is no substantial difference among score integration methods, and (5) our ranking diversification method was not effective.

Using a larger corpus for scoring subtopic candidate strings is one interesting future direction of this study. This is because it may allow us to measure the detailed difference of our proposed methods, to measure the effectiveness of our methods to rank less important subtopic strings, and to measure the effect of corpus size to our methods.

Another interesting future direction is to improve our diversified ranking method. In this paper, we completely removed blocks matching with already ranked strings. However, instead of such complete removal, we can reduce the scores of the blocks. This approach may be effective to score subtopics which ordinarily appear as sub-subtopics of other subtopics.

In this paper, we considered only re-ranking of already extracted subtopic candidate strings. However, of course, extraction of subtopic candidate strings is also an important step of subtopic mining. Therefore, extraction of hierarchical headings as subtopic candidate strings is also an important future direction of this study. However, to evaluate subtopic extraction methods, we need to expand the set of known subtopics because the NTCIR data set contains only a limited number of actual subtopics, which requires either some automatic method or a considerable amount of effort by human assesers.

Acknowledgment

This work was supported by JSPS KAKENHI Grant Number 13J06384 and 26540163.

References

1. Bah, A., Carterette, B., Chandar, P.: Udel @ NTCIR-11 IMine track. In: NTCIR (2014)
2. Bouchoucha, A., Nie, J., Liu, X.: Université de montréal at the NTCIR-11 IMine task. In: NTCIR (2014)
3. Carbonell, J., Goldstein, J.: The use of MMR, diversity-based reranking for reordering documents and producing summaries. In: SIGIR. pp. 335–336 (1998)

4. Clarke, C.L.A., Craswell, N., Voorhees, E.M.: Overview of the TREC 2012 web track. In: TREC (2012)
5. Das, S., Mitra, P., Giles, C.L.: Phrase pair classification for identifying subtopics. In: ECIR. pp. 489–493 (2012)
6. Dias, G., Cleuziou, G., Machado, D.: Informative polythetic hierarchical ephemeral clustering. In: WI. pp. 104–111 (2011)
7. Dou, Z., Hu, S., Chen, K., Song, R., Wen, J.R.: Multi-dimensional search result diversification. In: WSDM. pp. 475–484 (2011)
8. Dou, Z., Hu, S., Luo, Y., Song, R., Wen, J.R.: Finding dimensions for queries. In: CIKM. pp. 1311–1320 (2011)
9. Drosou, M., Pitoura, E.: Search result diversification. SIGMOD Rec. 39(1), 41–47 (2010)
10. He, J., Hollink, V., de Vries, A.: Combining implicit and explicit topic representations for result diversification. In: SIGIR. pp. 851–860 (2012)
11. Hu, Y., Qian, Y., Li, H., Jiang, D., Pei, J., Zheng, Q.: Mining query subtopics from search log data. In: SIGIR. pp. 305–314 (2012)
12. Jiang, D., Ng, W.: Mining web search topics with diverse spatiotemporal patterns. In: SIGIR. pp. 881–884 (2013)
13. Kim, S.J., Lee, J.H.: Subtopic mining using simple patterns and hierarchical structure of subtopic candidates from web documents. Inf. Process. Manage. 51(6), 773–785 (2015)
14. Liu, Y., Song, R., Zhang, M., Dou, Z., Yamamoto, T., Kato, M.P., Ohshima, H., Zhou, K.: Overview of the NTCIR-11 IMine task. In: NTCIR (2014)
15. Luo, C., Li, X., Khodzhaev, A., Chen, F., Xu, K., Cao, Y., Liu, Y., Zhang, M., Ma, S.: THUSAM at NTCIR-11 IMine task. In: NTCIR (2014)
16. Manabe, T., Tajima, K.: Extracting logical hierarchical structure of HTML documents based on headings. PVLDB 8(12), 1606–1617 (2015)
17. Manning, C.D., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S.J., McClosky, D.: The Stanford CoreNLP natural language processing toolkit. In: ACL. pp. 55–60 (2014)
18. Moreno, J.G., Dias, G.: HULTECH at the NTCIR-10 INTENT-2 task: Discovering user intents through search results clustering. In: NTCIR (2013)
19. Moreno, J.G., Dias, G.: HULTECH at the NTCIR-11 IMine task: Mining intents with continuous vector space models. In: NTCIR (2014)
20. Oyama, S., Tanaka, K.: Query modification by discovering topics from web page structures. In: APWeb. pp. 553–564 (2004)
21. Porter, M.F.: Readings in information retrieval. chap. An Algorithm for Suffix Stripping, pp. 313–316. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (1997)
22. Robertson, S.E., Walker, S.: Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval. In: SIGIR. pp. 232–241 (1994)
23. Sakai, T., Dou, Z., Yamamoto, T., Liu, Y., Zhang, M., Song, R.: Overview of the NTCIR-10 INTENT-2 task. In: NTCIR (2013)

24. Santos, R.L., Macdonald, C., Ounis, I.: Exploiting query reformulations for web search result diversification. In: WWW. pp. 881–890 (2010)
25. Song, R., Zhang, M., Sakai, T., Kato, M.P., Liu, Y., Sugimoto, M., Wang, Q., Orii, N.: Overview of the NTCIR-9 INTENT task. In: NTCIR (2011)
26. Strohman, T., Metzler, D., Turtle, H., Croft, W.: Indri: A language model-based search engine for complex queries. In: International Conference on Intelligent Analysis (2005)
27. Ullah, M.Z., Aono, M.: Query subtopic mining for search result diversification. In: ICAICTA. pp. 309–314 (2014)
28. Ullah, M.Z., Aono, M., Seddiqui, M.H.: SEM12 at the NTCIR-10 INTENT-2 english subtopic mining subtask. In: NTCIR (2013)
29. Wang, C., Danilevsky, M., Desai, N., Zhang, Y., Nguyen, P., Taula, T., Han, J.: A phrase mining framework for recursive construction of a topical hierarchy. In: KDD. pp. 437–445 (2013)
30. Wang, C.J., Lin, Y.W., Tsai, M.F., Chen, H.H.: Mining subtopics from different aspects for diversifying search results. Inf. Retr. 16(4), 452–483 (2013)
31. Wang, J., Tang, G., Xia, Y., Zhou, Q., Zheng, T.F., Hu, Q., Na, S., Huang, Y.: Understanding the query: THCIB and THUIS at NTCIR-10 intent task. In: NTCIR (2013)
32. Wang, Q., Qian, Y., Song, R., Dou, Z., Zhang, F., Sakai, T., Zheng, Q.: Mining subtopics from text fragments for a web query. Inf. Retr. 16(4), 484–503 (2013)
33. Xia, Y., Zhong, X., Tang, G., Wang, J., Zhou, Q., Zheng, T.F., Hu, Q., Na, S., Huang, Y.: Ranking search intents underlying a query. In: NLDB. pp. 266–271 (2013)
34. Xue, Y., Chen, F., Damien, A., Luo, C., Li, X., Huo, S., Zhang, M., Liu, Y., Ma, S.: THUIR at NTCIR-10 INTENT-2 task. In: NTCIR (2013)
35. Yamamoto, T., Kato, M.P., Ohshima, H., Tanaka, K.: KUIDL at the NTCIR-11 IMine task. In: NTCIR (2014)
36. Yamamoto, T., Liu, Y., Zhang, M., Dou, Z., Zhou, K., Markov, I., Kato, M.P., Ohshima, H., Fujita, S.: Overview of the NTCIR-12 IMine-2 task. In: NTCIR (2015)
37. Yu, H., Ren, F.: TUTA1 at the NTCIR-11 IMine task. In: NTCIR (2014)
38. Zeng, H.J., He, Q.C., Chen, Z., Ma, W.Y., Ma, J.: Learning to cluster web search results. In: SIGIR. pp. 210–217 (2004)
39. Zheng, W., Fang, H., Cheng, H., Wang, X.: Diversifying search results through pattern-based subtopic modeling. Int. J. Semant. Web Inf. Syst. 8(4), 37–56 (2012)
40. Zheng, W., Wang, X., Fang, H., Cheng, H.: An exploration of pattern-based subtopic modeling for search result diversification. In: JCDL. pp. 387–388 (2011)