Subtopic Ranking based on Hierarchical Headings

Tomohiro Manabe* and Keishi Tajima
Graduate School of Informatics, Kyoto University, Sakyo, Kyoto 606-8501, Japan

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

Abstract: We propose methods for generating diversified rankings of subtopics of keyword queries. Our methods are characterized by their awareness of hierarchical heading structure in documents. The structure consists of nested logical blocks with headings. Each heading concisely describes the topic of its corresponding block. Therefore, hierarchical headings in documents reflect the hierarchical topics referred to in the documents. Based on this idea, our methods score subtopic candidates based on matching between them and hierarchical headings in documents. They give higher scores to candidates matching hierarchical headings associated to more contents. To diversify the resulting rankings, every time our methods adopt a candidate with the best score, our methods exclude the blocks matching the candidate and re-score all remaining blocks and candidates. According to our evaluation result based on the NTCIR data set, our methods generated significantly better subtopic rankings than query completion results by major commercial search engines.

1 INTRODUCTION

Web search queries are sometimes ambiguous and/or referring to broad topics. To generate effective web page rankings for such queries, search result diversification techniques have been developed. Subtopic mining is one of the most promising approaches to search result diversification. Diversification methods based on subtopic mining first extract subtopic candidates of queries, then score and rank the candidates by their importance and their diversity from the others, and finally returns a few pages for each of the highly-ranked subtopic candidates. Because of the importance of subtopic mining, competitions for subtopic mining methods have been held as the NTCIR INTENT/IMine tasks subtopic mining subtasks (Liu et al., 2014; Sakai et al., 2013; Song et al., 2011).

In general, documents contain hierarchical heading structure reflecting their topic structure. Hierarchical headings structure consists of nested logical blocks and each block has its heading. A heading describes the topic of its associated block and the hierarchical descendant blocks of the block. Because of this feature of heading, hierarchical headings in documents reflect topic structure of the documents. For example, Figure 1 shows an example web page about computer programming (one of the NTCIR topics) 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 jobs, and the third level topics are computer programming school courses and degrees. Hierarchical heading structure of web pages are not obvious in general, but we have recently developed a method for extracting it (Manabe and Tajima, 2015).

In this paper, we propose methods to score hierarchical blocks in documents then rank subtopic candidates based on the scores of corresponding blocks. To the best of our knowledge, this is the first paper which discusses the use of detailed hierarchical heading structure of web pages in subtopic mining. Our basic idea is 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 the scores of the blocks in a corpus whose hierarchical headings describe the candidate subtopic. 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 with removing blocks matching with subtopics that have been already adopted. By this approach, the candidates

*Research Fellow of Japan Society for the Promotion of Science

Manabe, T. and Tajima, K.
Subtopic Ranking based on Hierarchical Headings.
In *Proceedings of the 12th International Conference on Web Information Systems and Technologies (WEBIST 2016) - Volume 2*, pages 121-130
ISBN: 978-989-758-186-1

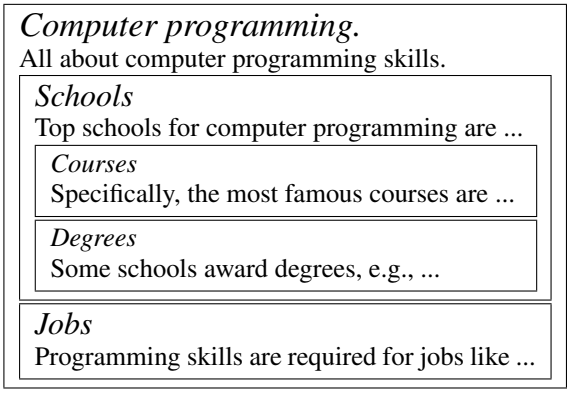


Figure 1: Example web page with hierarchical heading structure. Each rectangle encloses block and each emphasized text is heading. Some long texts are replaced by dots.

matching with the blocks which also match with the already-adopted subtopics 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 the publicly available NTCIR data set and compare the results with the baselines generated by commercial web search engines. Lastly, Section 6 concludes this paper.

2 DEFINITIONS

In this section, we clarify the definitions of our research targets, 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 as quoted below (Sakai et al., 2013).

> 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 topic behind an original query. In INTENT-2 and in this paper, a *query* means a keyword query composed of an array of words.

The overview paper of the task in NTCIR-10 lists some example subtopic strings (Sakai et al., 2013). If the original query is “harry potter”, “harry potter philosophers stone movie” is a true subtopic string that specializes the original query. On the other hand, “harry potter hp” is not a subtopic string because it neither specializes nor disambiguates the original query. If the original query is “office”, “office workplace” is a subtopic string that disambiguates the original query, but “office office” is not. Note that true subtopic strings may not include the original queries. For example, “aliens vs predators” is a true subtopic string of the original query “avp”.

2.2 Definition of Heading Structure

For ranking of subtopics, we use hierarchical heading structure of documents. We define the structure and its components as summarized below (Manabe and Tajima, 2015).

Heading: A *heading* is a highly summarized description of the topic of a part of a document.

Block: As explained above, a heading is associated with a *block*, a clearly specified part of a document. We consider neither a block that consists only of its heading nor a block without its heading. An entire document is also a block because it is clearly specified 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. All blocks in a document form a hierarchical heading structure whose root is the *root block* representing the entire document.

3 RELATED WORK

Generally, a term *topic* has two meanings in informatics (He et al., 2012). One is an implicit topic represented by a (fuzzy) set of terms (Jiang and Ng, 2013; Hu et al., 2012), and the other is an explicit topic represented by a short string like a keyword query. Our research target is explicit topics. In particular, we focus on subtopics of the topics behind the keyword queries input by users. For mining such subtopics, we need four component technologies. They are namely subtopic candidate extraction, extraction of their features, 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 recommendation/suggestion/completion by search engines generates many related queries of

the original queries. They are very popular resources of subtopic candidate strings (Luo et al., 2014; Yu and Ren, 2014; Ullah et al., 2013; Xue et al., 2013; Ullah and Aono, 2014; Wang et al., 2013b; Xia et al., 2013), and the snapshots of them for the NTCIR-10 INTENT-2 task (Sakai et al., 2013) is publicly available. We also adopted them as baseline subtopics. Google Insights and Google keywords generator are similar services (Xue et al., 2013). Raw query logs of search engines (Luo et al., 2014; Bouchoucha et al., 2014; Ullah et al., 2013; Wang et al., 2013b; Xia et al., 2013) must also be useful.

Disambiguation pages in Wikipedia contain multiple subtopics of many ambiguous article titles of Wikipedia, and are very well-organized by humans. Therefore, they are also a very popular resource of subtopic candidate strings (Wang et al., 2013b; Xia et al., 2013; Luo et al., 2014; Yu and Ren, 2014; Xue et al., 2013; Wang et al., 2013b; Xia et al., 2013). Redirect pages and tables of contents in Wikipedia must also be useful (Xia et al., 2013).

Of course, search result documents themselves can be a resource of subtopic candidate strings. Methods based on frequently occurring words (Yamamoto et al., 2014; Oyama and Tanaka, 2004; Wang et al., 2013b; Wang et al., 2013a; Zheng et al., 2012; Zheng et al., 2011), words frequently co-occurring with query keywords (Wang et al., 2013d), pseudo-relevance feedback (Bouchoucha et al., 2014), syntactic patterns (Kim and Lee, 2015), search result summaries (Xue et al., 2013) have been proposed.

Titles (Oyama and Tanaka, 2004; Yamamoto et al., 2014), anchor texts of in-links (Xue et al., 2013; He et al., 2012), and explicitly tagged top-level headings (H1 nodes) of HTML documents (Xue et al., 2013) 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. 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 (Dou et al., 2011). 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 (Bah et al., 2014; Ullah et al., 2013).

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 (Kim and Lee, 2015; Yamamoto et al., 2014; Wang et al., 2013d; Zheng et al., 2012; Das et al., 2012). TF means the number of its occurrences in a document, and DF means the number of documents that 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 (Yamamoto et al., 2014; Xue et al., 2013).

Similarity between subtopic candidate strings and their search result documents or their original queries is a popular feature (Luo et al., 2014; Moreno and Dias, 2014; Zheng et al., 2012; Das et al., 2012).

The 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 (Kim and Lee, 2015), and the distinctness entropy of subtopic candidate strings measures the distinctness among the document sets that the strings retrieved (Zeng et al., 2004; Kim and Lee, 2015).

The SEM group at NTCIR-10 used the co-occurrence of subtopic candidate strings in query logs and the edit distance between the strings and their original queries (Ullah et al., 2013).

Query-independent features like readability of subtopic candidate strings are also useful (Ullah et al., 2013; Wang et al., 2013d).

3.3 Subtopic Ranking

Subtopic ranking is indispensable for filtering out noises and for ranking subtopic strings by the probability that they are the query intent. The simplest way is to sort them in order of linear combination of features. However, as in document ranking, more sophisticated functions, e.g., TFIDF (TF over DF) and BM25 (Robertson and Walker, 1994), are also used (Kim and Lee, 2015; Wang et al., 2013d; Wang et al., 2013b; Ullah et al., 2013).

Many methods assign different weights for different sources of subtopic candidate strings (Luo et al., 2014; Xue et al., 2013). For example, the THUIR group at 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 (Xue et al., 2013).

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 (Ullah and Aono, 2014).

It is notable that the THUSAM group at NTCIR-12 adopted a variant of learning-to-rank methods that

are state-of-the-arts methods for document ranking (Luo et al., 2014).

3.4 Subtopic Diversification

One important application of subtopic mining methods is search result diversification. Therefore, diversity of subtopic rankings is also important.

One promising way to diversify subtopic rankings is subtopic clustering and extraction of the median subtopics of each cluster (Yamamoto et al., 2014; Yu and Ren, 2014; Xue et al., 2013; Wang et al., 2013b; Wang et al., 2013a; Xia et al., 2013). The K-means (Yamamoto et al., 2014), affinity propagation (Yu and Ren, 2014; Xia et al., 2013), a variant of K-medoids (Xue et al., 2013) 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 (Wang et al., 2013c).

The Hierarchical InfoSimba-based Global K-means (HISGK-means) algorithm clusters search result snippets then labels each cluster (Moreno and Dias, 2013; Dias et al., 2011). 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 (Luo et al., 2014; Moreno and Dias, 2014). In word embedding models, we can *subtract* subtopic candidate strings from their original query. Based on this idea, the HULTECH group at NTCIR-11 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 (Moreno and Dias, 2014).

The maximal marginal relevance (MMR) framework also concatenate items into rankings one-by-one (Carbonell and Goldstein, 1998). In each iteration, the MMR framework selects the item with the best balance of the score and dissimilarity to the already ranked items. Of course, it is useful for subtopic diversification (Ullah and Aono, 2014).

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, as in our method.

4 SUBTOPIC RANKING BASED ON HIERARCHICAL HEADING STRUCTURE

In this section, we propose scoring and ranking methods for subtopic strings. Our proposed methods are based on matching between the subtopic strings and hierarchical heading structure of documents in a corpus. We regard that a subtopic string *matches* a block iff all the words in the subtopic string appear either in the heading of the block or in the headings of its ancestor blocks. For example, a subtopic string "computer programming degrees" matches the "degrees" block in Figure 1. 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 descendants. Formally, the score of a pair of a subtopic string s and a document d is:

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

where b is each block in d, $\mathrm{match}(s,b)$ is 1 iff s matches b and does not match any ancestor block of b and is 0 otherwise. $\mathrm{blockScore}(b)$ is the score of b.

Hereafter in this section, we first discuss the definition of $\mathrm{blockScore}(b)$, then discuss integration of subtopic scores on multiple documents, and finally discuss ranking of multiple subtopics into a diversified ranking.

4.1 Subtopic Scoring on a Single Page

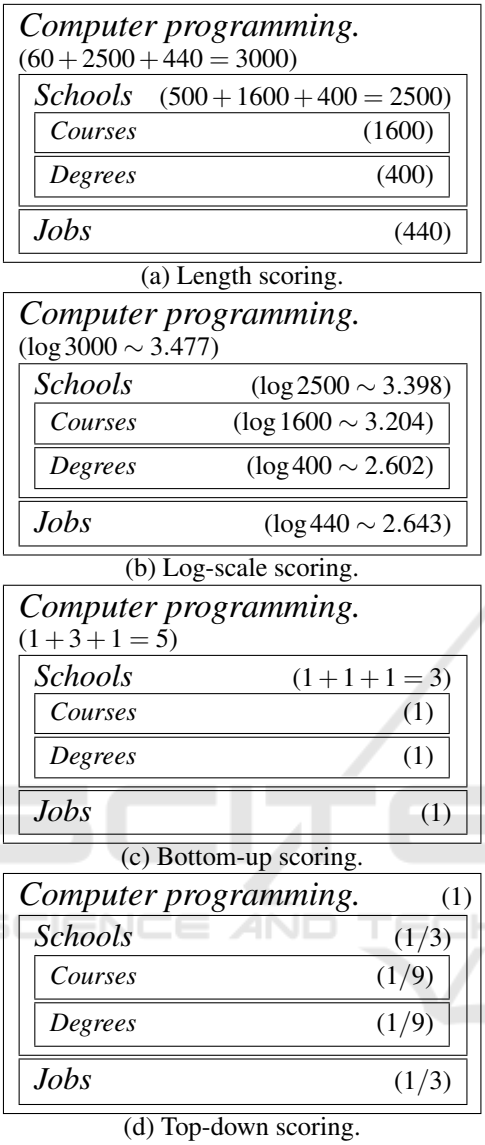
First, we propose four scoring methods for blocks.

4.1.1 Scoring by Content Length

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

$$\mathrm{blockScore}(b) = \mathrm{length}(b)$$

where $\mathrm{length}(b)$ 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.



(a) Length scoring.

(b) Log-scale scoring.

(c) Bottom-up scoring.

(d) Top-down scoring.

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

In Figure 2, the scores of the blocks are in parentheses and non-heading components of the blocks are omitted.

4.1.2 Scoring by Log-Scaled Content Length

As the relevance of a document to a query is assumed not to be direct proportional to the number of query keyword occurrences in the document (Robertson and Walker, 1994), 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:

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

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

4.1.3 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:

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

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.

4.1.4 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 the original block may have meaningful contents besides its child blocks, we also assign the same importance to the contents. The score of a block b is:

$$\text{blockScore}(b) = \begin{cases} \frac{\text{blockScore}(p)}{1+|p|} & \text{if } b \text{ has a parent block } p \\ 1 & \text{otherwise} \end{cases}$$

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 string on multiple documents.

4.2.1 Simple Summation

The simplest way to integrate the scores for multiple pages 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:

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

We call this method *summation* integration.

4.2.2 Page-based Integration

In summation integration, documents of more length or including more blocks have more chance to contribute to $\text{score}(s,D)$. However, if we assume each document is equally important, the scaling of $\text{docScore}(s,d)$ defined below may be useful:

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

where $\text{root}(d)$ is the root block in d, i.e., the block representing entire d. Because we score each block considering its hierarchical descendant blocks, $\text{blockScore}(b)$ takes its maximum value in a document when b is the root block of the document, and $\text{docScore}(s,d)$ takes its maximum value when s matches the root block of d. Therefore, this division by $\text{blockScore}(\text{root}(d))$ scales the $\text{docScore}(s,d)$ to $[0,1]$. We call this method *page-based* integration.

Note that there is no difference between summation and page-based integration when we use top-down scoring because $\text{blockScore}(b)$ in top-down scoring is already scaled to $[0,1]$.

4.2.3 Domain-based Integration

Some authors may split a topic into multiple documents in a *domain*, e.g., a set of web pages whose URL have the same domain, instead of multiple blocks. 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 that appear in the corpus. Each domain $\delta \in \Delta$ is a subset of the corpus D, and $\bigcup_{\delta \in \Delta} \delta = D$. The new integration function is:

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

We call this method *domain-based* integration.

4.2.4 Combination Integration

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

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

Computer programming.
($\log 500 \sim 2.699$)

Jobs	($\log 440 \sim 2.643$)

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

We call this *combination* integration.

4.3 Diversifying Subtopic 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.

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 of subtopic ranking. Our idea for diversification is that if a block matches a subtopic string that is already ranked in the ranking, the topic of the block is already referred to by the subtopic string, and therefore, even if the block matches some other remaining subtopic strings, the block should not contribute to the score of the subtopic 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 string on a document set then put only the string with the best score into the resulting ranking. Second, we remove all the blocks matching with the string from the corpus. Third, we re-score the remaining subtopic 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 subtopic strings 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 in Figure 2b, the ranks of the strings are in the order above because the strings match the "Schools" (score: 3.398), "Courses" (score: 3.204), and "Jobs" (score: 2.643) blocks, respectively. On the other hand, if we rank the strings by diversified ranking method, "computer programming jobs" achieves the second rank because after "computer programming school" is ranked first, its matching block "School" including its descendant blocks is removed from the recalculation of the scores. Then the score of "computer programming course" in this page becomes 0 because

the block referring to the subtopic in this page has already matched the higher ranked subtopic "computer programming school".

5 EVALUATION

In this section, we evaluate and compare the baselines and 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 integration when we use top-down scoring as discussed in Section 4.2.2. Therefore, we compare 28 proposed methods in total.

5.1 Evaluation Methodology

Because we do not discuss extraction methods of subtopic candidate strings, we evaluate our ranking methods by re-ranking the baseline subtopic rankings.

We use the official data set (including baselines) and evaluation measures of the NTCIR-10 INTENT-2 task subtopic mining subtask (Sakai et al., 2013). This is because the dataset of the latest NTCIR-12 IMine-2 task is not available yet, and because first-level and second-level subtopics are distinguished in the second-latest NTCIR-11 IMine task 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[2].

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 (or *intent*). Multiple subtopic strings may refer to the same intent, but a string refers to one intent 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:

$$\text{iRec@10} = |I'|/|I|$$

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 the recall and diversity of subtopics in rankings. The definition of the D-nDCG measure for a ranking of maximum 10 strings is:

$$\begin{aligned} \text{DnDCG@10} &= \frac{\text{DDCG@10}}{\text{ideal DDCG@10}} \\ \text{where DDCG@10} &= \sum_{r=1}^{10} \frac{\sum_i Pr(i|q)g_i(r)}{\log(r+1)} \end{aligned}$$

where r is a rank, $Pr(i|q)$ is the intent probability of a 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 rankings.

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

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

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.

An official evaluation tool is available online[3].

5.2 Data Set

The details of the data set is as follows.

Queries: We used 50 keyword queries in the NTCIR data set that are also used in the Text Retrieval Conference (TREC) 2012 Web track (Clarke et al., 2012).

Document Sets: We used the baseline document rankings generated by default scoring of Indri search engine (including query expansion based on pseudo-relevance feedback) and Waterloo spam filter for TREC 2012 Web track. The baseline rankings are available online[4]. Each ranking consists of 131–837 web pages from ClueWeb09B for a query. The ClueWeb09B 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 collection is also available at distribution cost[5].

Baseline Results: There are the snapshots of query completion/suggestion results by commercial search engines prepared for the NTCIR-10 INTENT-2 task. We used the query completion results by Google and Yahoo because they achieved the best I-rec and D-nDCG scores respectively among the baselines (Sakai et al., 2013). Because the both results contain only 10 strings at most for each query, re-ranking of them do not affect the I-rec scores. Therefore, we also used our *merged* baseline result which is generated by merging four baseline query completion/suggestion results and sorting them in "dictionary sort" (Sakai et al., 2013). Because the meaning of dictionary sort

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

[3]http://research.nii.ac.jp/ntcir/tools/ntcireval-en.html

[4]https://github.com/trec-web/trec-web-2014

[5]http://www.lemurproject.org/clueweb09/

is ambiguous, we could not reproduce their evaluation result. We merged the results, decapitalized the strings, removed duplicated strings, and sort the remaining strings in byte order in UTF-8 to generate our merged baseline result.

Known Intents and Intent Probabilities: The known intents, subtopic strings referring to them, and their intent probabilities are manually prepared for the subtask (Sakai et al., 2013). All the true subtopic strings in the baseline results must be in this data according to their annotation process.

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 (Manabe and Tajima, 2015). It extracts each heading and block in pages as an array of adjoining sibling DOM nodes. For evaluation, we used the reference implementation of HEPS 1.0.0[6].

Text Contents of Headings and Blocks: We used the URL and 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 (Manabe and Tajima, 2015). 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.

Content Length: For length and log-scaled scoring, we used the number of characters in their raw strings as their length.

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

Matching between Subtopic Strings and Headings: Before matching subtopic candidates and hierarchical headings, we performed basic preprocessing for retrieval, e.g., tokenization, stop word filtering, and stemming, for both strings. All URLs were split by any non-word characters, and the other strings are tokenized by Stanford CoreNLP toolkit (Manning et al., 2014). All tokens were decapitalized, filtered out 33 default stop words of the Lucene library[7], then stemmed by the Porter stemmer (Porter, 1997).

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

[6]https://github.com/tmanabe/HEPS

[7]http://lucene.apache.org/

Table 1: D-nDCG score comparison with query completion by Google. Top-5 proposed methods are listed. For all methods and baseline, I-rec = 0.3841.

Scoring	Integration	Ranking	Score
log-scale	domain-based	uniform	**.4502**
log-scale	combination	uniform	.4501
log-scale	domain-based	diversified	.4487
log-scale	combination	diversified	.4485
bottom-up	page-based	diversified	.4479
Query completion result by Google			.3735

Table 2: D-nDCG score comparison with query completion by Yahoo. Top-5 proposed methods are listed. For all methods and baseline, I-rec = 0.3815.

Scoring	Integration	Ranking	Score
log-scale	page-based	diversified	**.4617**
bottom-up	domain-based	diversified	.4609
log-scale	page-based	uniform	.4608
log-scale	summation	diversified	.4601
length	domain-based	diversified	.4587
Query completion result by Yahoo			.3829

Ties: 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 ranking.

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 by Google, and Table 2 shows the D-nDCG scores achieved by each method when they re-rank the query completion by Yahoo. In Table 1 and 2, top-5 proposed 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, top-5 proposed methods are listed in descending order of their D♯-nDCG scores. In this comparison, the log-scale/summation/uniform method achieved the best scores in all the measures among all the proposed methods including ones omitted from Table 3.

5.5 Discussion

In all comparisons, our proposed methods achieved the better scores than the baseline results. This is not because we proposed a number of methods and one of them achieved a better score than each baseline result by chance. For example, let us focus on the log-scale/page-based/diversified method which achieved the best D♯-nDCG score throughout this experiment

Table 3: Comparison with our merged baseline result. Top-5 methods are listed in descending order of their D♯-nDCG scores.

Scoring	Integration	Ranking	I-rec	D-nDCG	D♯-nDCG
log-scale	summation	uniform	**.4009**	**.3997**	**.4003**
log-scale	page-based	uniform	.3986	.3981	.3984
length	summation	uniform	.3974	.3945	.3959
log-scale	combination	uniform	.3956	.3921	.3939
log-scale	domain-based	uniform	.3956	.3913	.3934
Our merged baseline result			.3310	.3066	.3188

by reranking the result by Yahoo. The method also achieved a better D-nDCG score (0.4470) than the result by Google and better I-rec, D-nDCG, and D♯-nDCG scores (0.3840, 0.3695, and 0.3768, respectively) than the merged result. 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 our proposed subtopic ranking 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.

5.5.1 Comparison of Block Scoring Methods

Log-scale scoring achieved the best scores in all the three comparisons. This fact may suggest that 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 Yahoo result reranked 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.

5.5.2 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 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.

5.5.3 Effect of Diversified Ranking Method

Because I-rec measures the diversity of rankings, we focus on the I-rec score comparison with our merged result (Table 3). No method with diversified ranking achieved the top-5 scores. The top-down/combination/*diversified* method achieved the best I-rec score (0.3869) among the methods with diversified ranking. The I-rec score difference between the method and the best log-scale/summation/*uniform* method was not statistically significant ($p = 0.2759$). The I-rec score difference between the best method and the log-scale/summation/*diversified* method was also not statistically significant ($p = 0.1028$). They show that our proposed ranking diversification method did neither improve nor worsen resulting rankings.

6 CONCLUSION

We proposed subtopic ranking methods based on the ideas that hierarchical headings in a document reflect the topic structure of the document and that the length of contents referring to a topic reflects the importance of the topic. Our methods rank candidate subtopics based on the blocks whose hierarchical headings match the subtopic candidate strings. We evaluated our methods by using the publicly available NTCIR data set. The results indicated (1) our methods significantly improved the baseline rankings by commercial search engines, (2) log-scale scoring seems effective and robust, (3) there is no substantial difference among score integration methods, and (4) our ranking diversification method was not effective.

ACKNOWLEDGMENT

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

REFERENCES

Bah, A., Carterette, B., and Chandar, P. (2014). Udel @ NTCIR-11 IMine track. In *NTCIR*.

Bouchoucha, A., Nie, J., and Liu, X. (2014). Université de montréal at the NTCIR-11 IMine task. In *NTCIR*.

Carbonell, J. and Goldstein, J. (1998). The use of MMR, diversity-based reranking for reordering documents and producing summaries. In *SIGIR*, pages 335–336.

Clarke, C. L. A., Craswell, N., and Voorhees, E. M. (2012). Overview of the TREC 2012 web track. In *TREC*.

Das, S., Mitra, P., and Giles, C. L. (2012). Phrase pair classification for identifying subtopics. In *ECIR*, pages 489–493.

Dias, G., Cleuziou, G., and Machado, D. (2011). Informative polythetic hierarchical ephemeral clustering. In *WI*, pages 104–111.

Dou, Z., Hu, S., Luo, Y., Song, R., and Wen, J.-R. (2011). Finding dimensions for queries. In *CIKM*, pages 1311–1320.

He, J., Hollink, V., and de Vries, A. (2012). Combining implicit and explicit topic representations for result diversification. In *SIGIR*, pages 851–860.

Hu, Y., Qian, Y., Li, H., Jiang, D., Pei, J., and Zheng, Q. (2012). Mining query subtopics from search log data. In *SIGIR*, pages 305–314.

Jiang, D. and Ng, W. (2013). Mining web search topics with diverse spatiotemporal patterns. In *SIGIR*, pages 881–884.

Kim, S.-J. and Lee, J.-H. (2015). Subtopic mining using simple patterns and hierarchical structure of subtopic candidates from web documents. *Inf. Process. Manage.*, 51(6):773–785.

Liu, Y., Song, R., Zhang, M., Dou, Z., Yamamoto, T., Kato, M. P., Ohshima, H., and Zhou, K. (2014). Overview of the NTCIR-11 IMine task. In *NTCIR*.

Luo, C., Li, X., Khodzhaev, A., Chen, F., Xu, K., Cao, Y., Liu, Y., Zhang, M., and Ma, S. (2014). THUSAM at NTCIR-11 IMine task. In *NTCIR*.

Manabe, T. and Tajima, K. (2015). Extracting logical hierarchical structure of HTML documents based on headings. *PVLDB*, 8(12):1606–1617.

Manning, C. D., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S. J., and McClosky, D. (2014). The Stanford CoreNLP natural language processing toolkit. In *ACL*, pages 55–60.

Moreno, J. G. and Dias, G. (2013). HULTECH at the NTCIR-10 INTENT-2 task: Discovering user intents through search results clustering. In *NTCIR*.

Moreno, J. G. and Dias, G. (2014). HULTECH at the NTCIR-11 IMine task: Mining intents with continuous vector space models. In *NTCIR*.

Oyama, S. and Tanaka, K. (2004). Query modification by discovering topics from web page structures. In *APWeb*, pages 553–564.

Porter, M. F. (1997). Readings in information retrieval. chapter An Algorithm for Suffix Stripping, pages 313–316. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

Robertson, S. E. and Walker, S. (1994). Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval. In *SIGIR*, pages 232–241.

Sakai, T., Dou, Z., Yamamoto, T., Liu, Y., Zhang, M., and Song, R. (2013). Overview of the NTCIR-10 INTENT-2 task. In *NTCIR*.

Song, R., Zhang, M., Sakai, T., Kato, M. P., Liu, Y., Sugimoto, M., Wang, Q., and Orii, N. (2011). Overview of the NTCIR-9 INTENT task. In *NTCIR*.

Ullah, M. Z. and Aono, M. (2014). Query subtopic mining for search result diversification. In *ICAICTA*, pages 309–314.

Ullah, M. Z., Aono, M., and Seddiqui, M. H. (2013). SEM12 at the NTCIR-10 INTENT-2 english subtopic mining subtask. In *NTCIR*.

Wang, C., Danilevsky, M., Desai, N., Zhang, Y., Nguyen, P., Taula, T., and Han, J. (2013a). A phrase mining framework for recursive construction of a topical hierarchy. In *KDD*, pages 437–445.

Wang, C.-J., Lin, Y.-W., Tsai, M.-F., and Chen, H.-H. (2013b). Mining subtopics from different aspects for diversifying search results. *Inf. Retr.*, 16(4):452–483.

Wang, J., Tang, G., Xia, Y., Zhou, Q., Zheng, T. F., Hu, Q., Na, S., and Huang, Y. (2013c). Understanding the query: THCIB and THUIS at NTCIR-10 intent task. In *NTCIR*.

Wang, Q., Qian, Y., Song, R., Dou, Z., Zhang, F., Sakai, T., and Zheng, Q. (2013d). Mining subtopics from text fragments for a web query. *Inf. Retr.*, 16(4):484–503.

Xia, Y., Zhong, X., Tang, G., Wang, J., Zhou, Q., Zheng, T. F., Hu, Q., Na, S., and Huang, Y. (2013). Ranking search intents underlying a query. In *NLDB*, pages 266–271.

Xue, Y., Chen, F., Damien, A., Luo, C., Li, X., Huo, S., Zhang, M., Liu, Y., and Ma, S. (2013). THUIR at NTCIR-10 INTENT-2 task. In *NTCIR*.

Yamamoto, T., Kato, M. P., Ohshima, H., and Tanaka, K. (2014). KUIDL at the NTCIR-11 IMine task. In *NTCIR*.

Yu, H. and Ren, F. (2014). TUTA1 at the NTCIR-11 IMine task. In *NTCIR*.

Zeng, H.-J., He, Q.-C., Chen, Z., Ma, W.-Y., and Ma, J. (2004). Learning to cluster web search results. In *SIGIR*, pages 210–217.

Zheng, W., Fang, H., Cheng, H., and Wang, X. (2012). Diversifying search results through pattern-based subtopic modeling. *Int. J. Semant. Web Inf. Syst.*, 8(4):37–56.

Zheng, W., Wang, X., Fang, H., and Cheng, H. (2011). An exploration of pattern-based subtopic modeling for search result diversification. In *JCDL*, pages 387–388.