Heading-Aware Snippet Generation for Web Search

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 heading-aware methods of generating search result snippets of web pages. A heading is a brief description of the topic of its associated sentences. Some existing methods give priority to sentences containing many words that also appear in headings when selecting sentences to be included in snippets with limited length. However, according to our observation, words in heading are very often omitted from their associated sentences because readers can understand the topic of the sentences by reading their heading. To score sentences considering such omission, our methods count keyword occurrences in their headings as well as in the sentences themselves. Our evaluation result indicated that our methods were effective only for queries with clear intents or containing four or more keywords. To discuss the statistical significance of the result, another evaluation with more queries is needed.

Keywords: Snippet generation, Query-biased summarization, Web search result snippets, Heading Structure

1 Introduction

Most web pages contain hierarchical heading structure [11]. The structure is composed of nested logical blocks and each block is associated with a heading that briefly describes the topic of the block. Because of this feature of headings, to fully understand sentences in web pages, readers should first read the contextual headings of the sentences. The contextual headings (or merely headings) of a sentence are the headings associated with either the block containing the sentence or its hierarchical ancestor blocks. Therefore, the contextual heading words (or merely heading words), i.e. the words in the contextual headings, are important for understanding their associated sentences.

For this reason, there have been several studies on heading-aware snippet generation [13, 17]. These methods assign higher scores to headings themselves [17] or sentences containing their heading words [13]. However, contextual heading words are very often omitted from their associated sentences because human readers can recognize the topic of the sentences by reading the headings first.

* Research Fellow of Japan Society for the Promotion of Science

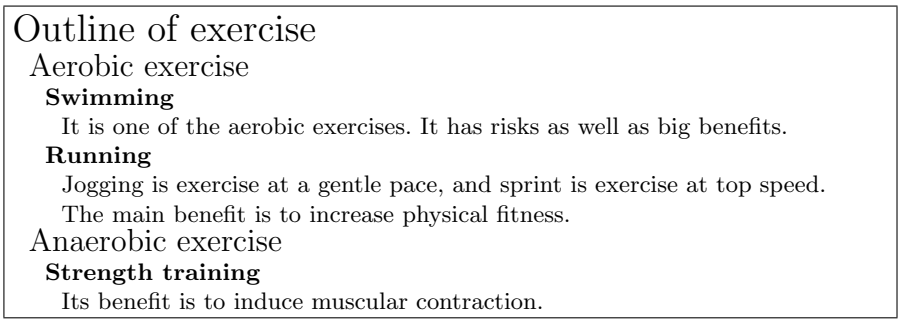

Fig. 1. Example web page with hierarchical heading structure.

For example, in the example page in Figure 1, by the sentence "It has risks as well as big benefits", the author is writing about swimming without the word swimming. For a query "swimming risks", the existing methods cannot assign higher relevance scores for such sentences.

To solve the problem, we develop a new method of heading-aware snippet generation that takes the omission of heading words into account. Our method assign higher scores to sentences that either include query keywords within themselves or have contextual headings including query keywords. Our new approach does not conflict with the existing approach that uses heading-word occurrences in sentences. Therefore, we also consider another method which combines the two types of evidences, namely heading-word occurrences in sentences themselves and query keyword occurrences in the contextual headings of sentences.

2 Related Work

Generally, snippet generation methods uses some types of important words and document fragments. Almost all methods count the occurrences of query keywords. Additionally, some methods use pseudo relevance feedback to expand queries and obtain more keywords [8, 19]. Frequently occurring words in a page may also be important for the page [13, 17, 19]. The first paragraph of a page [17] or the first sentence of a paragraph [13] may also be important. As listed above, most summarization methods do not focus on heading words and headings.

As explained in Section 1, two heading-aware summarization methods exist. The method by Tombros and Sanderson regards headings as important sentences and assigns higher scores to headings than to other sentences [17]. However, as discussed in Section 1, headings are also important for scoring other sentences. The method by Pembe and Güngör counts heading-word occurrences in sentences to score the sentences [13]. However, as also discussed in Section 1, their method does not take the omission of heading words into account.

Some snippet generation methods focus on the locations of the occurrences of query keywords. Some methods count the occurrences in document titles, which are a type of headings [17, 19]. The method by Zhang et al. distinguishes

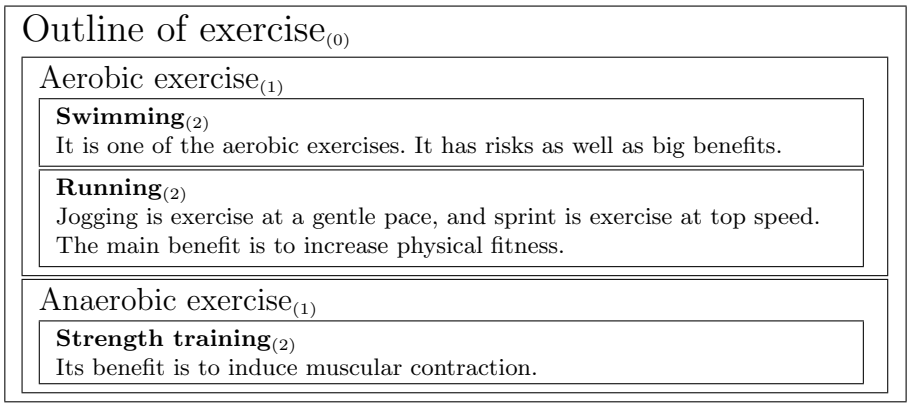

Fig. 2. Hierarchical heading structure of page in Figure 1. Each rectangle encloses block, each text with subscript is heading and each subscript number represents depth of block in the hierarchy.

attribute names, which are also a type of headings [20]. These methods, however, do not count query keyword occurrences in general headings.

Outside the field of web search, many snippet generation methods for XML documents are based on XML element retrieval [8, 19], and many XML element retrieval methods take the hierarchical ancestors of elements into account [2, 3]. However, unlike our methods, most XML element retrieval methods do not distinguish headings from other components of elements. The BM25E function for element scoring distinguish headings from other components [10]. However, the application of the function to snippet generation has not been discussed.

3 Heading Structure Extraction

Hierarchical heading structure of web pages is not obvious. In this section, we introduce an outline of HEPS, our previously proposed method for extracting the implicit hierarchical heading structure from HTML web pages [11]. Throughout this paper, we assume that the hierarchical heading structure of web pages are already extracted by this method. See our previous paper for the detailed design decisions and evaluation results of HEPS itself [11].

First we define the hierarchical heading structure and its components.

Heading: In our definition, a *heading* is a highly summarized description of the topic of a part of a web page.

Block: As explained above, a heading is associated with a *block*, a clearly specified region in a web page. We consider neither a block that consists of its heading only nor a block without its heading. A whole web page 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 page form a hierarchical heading structure whose root is the block representing the entire page.

In Figure 2, we show the hierarchical heading structure in the example page in Figure 1. Each block (including the page) is enclosed by a rectangle, and its heading is associated with a subscript representing its depth in the hierarchy.

The HEPS method involves pre-processing and three main steps. In the first main step, it classifies DOM nodes into sets of nodes sharing the same visual style (e.g., font size and font weight). Second, it sorts the sets in descending order of visual significance of their elements. Third, it determines the actual heading set of the highest significance and divides the page into blocks. The third step is recursively repeated to divide a page into nested blocks.

4 Snippet Generation Methods

In this section, we explain four snippet generation methods for web search.

4.1 Basic Snippet Generation Method

Generally, the quality of document summaries relies on three factors [1]. The *readability* of a summary is how easy it is for humans to read [5, 7], its *representativeness* is how well it represents the contents of the original document [9], and its *judgeability* is to what extent it helps users to judge the relevance of the original document to the users' informational needs [9]. Among the three factors, judgeability is the most important for search result snippets.

Basically, search result snippets are generated from web pages by search systems in three steps as described below [13, 17, 19]. First, the system splits the page into text fragments. To generate readable snippets, many systems split it into semantically coherent fragments such as sentences. Second, the system scores the fragments based on the numbers of the occurrences of important words in the fragments. The occurrences of the query keywords directly indicate the relevance of the original page to the users' intent behind the query. Therefore, almost all systems take keyword occurrences into account for higher judgeability. On the other hand, other important words (see Section 2) in a document represent the contents of the original document better than other words. Therefore, many systems take important-word occurrences into account for higher representativeness. Third, the system selects the top-ranked sentences into the summary. In this step, the system selects the sentences in descending order of their scores until the length of the summary reaches the limit. Our *baseline* method also consists of these three steps. The method is described as below.

Input: A web page with its DOM tree structure. Note that we consider only documents in English throughout this paper. This is merely because we use some language-dependent libraries for sentence segmentation and stemming. Note that our heading-aware sentence scoring methods are language-independent.

Sentence Segmentation: First, the method extracts the text contents of the page, then segments the contents into sentences. As the text contents of a page, we extract the text contents of all text and IMG (image) nodes under the BODY (content body) node of the page, and concatenate them in the document order.

Outline of exercise
... It has risks as well as big benefits. ... Running ... The main benefit is to increase physical fitness. ... Its benefit is to induce muscular contraction.

Fig. 3. Example baseline snippet for example query "benefits running".

As the text contents of IMG nodes, we extract their alternate text. From IMG nodes without alternate text, we extract the URLs of the images. We split the text contents of the page into sentences by the Stanford CoreNLP toolkit [12].
Sentence Scoring: We score the sentences based on the number of keyword occurrences in them by a variant of the BM25 function [15]. The function calculates the score of a sentence s for keyword query q by the following formula:

$$\text{score}(q, s) = \sum_{\kappa \in q} \frac{\text{weight}(\kappa, s)}{k_1 + \text{weight}(\kappa, s)} \log \frac{N - \text{sf}(\kappa) + 0.5}{\text{sf}(\kappa) + 0.5} \tag{1}$$

where κ is a keyword in q, k_1 is a parameter to modify the scaling of occurrence frequency, N is the number of all sentences, and $\text{sf}(\kappa)$ is the number of sentences containing κ in the page. The $\text{weight}(\kappa, s)$ is defined as

$$\text{weight}(\kappa, s) = \frac{\text{occurs}(\kappa, s)}{\left((1 - b) + b \cdot \frac{\text{length}(s)}{\text{avgLength}}\right)} \tag{2}$$

where $\text{occurs}(\kappa, s)$ is the number of occurrences of κ in s, b is the parameter to modify the strength of length normalization, $\text{length}(s)$ is the length of s in number of words, and avgLength is the average length of sentences in the page. We count $\text{occurs}(\kappa, s)$ after the basic pre-processing, i.e. stemming by the Porter stemming algorithm [14] and removal of 33 default stop words of Apache Lucene.
Sentence Selection: To select the sentences, we simply scan the sentences in descending order of their scores, and if there still remains the space to include the sentence into the snippets, we include it. We can also adopt advanced methods for the selection, such as Maximal Marginal Relevance [4, 20], however, we adopt this simple method because this step is not the main topic of this paper.
Output: The generated snippet and the title of the input page. If there is no page title specified, we output the page URL. Figure 3 is an example output.

4.2 Occurrences of Heading Words in Sentences

Heading words are important to represent their associated blocks because the words are selected by the authors to describe the topics of the blocks briefly. As discussed in Section 1, we consider the contextual headings and heading words of sentences. For example, the heading words of the sentence "It is one of the aerobic exercises" in Figure 2 are outline, of, exercise, aerobic, and swimming.

One promising way to generate representative snippets is to extract sentences containing many occurrences of their contextual heading words. Pembe

Outline of exercise
> **Aerobic exercise** > **Swimming**
It is one of the aerobic exercises. ...
> **Aerobic exercise** > **Running**
Jogging is exercise at a gentle pace, and sprint is exercise at top speed. The main benefit is to increase physical fitness.

Fig. 4. Example heading-aware snippet for example query "benefits running".

and Güngör [13] proposed such a method. We also use this idea for our *existing* method, which is based on summation of the BM25 scores for two types of words, namely query keywords and heading words. However, a weighted summation of BM25 scores produces a worse ranking in case that they count occurrences of the same words [16]. Therefore, we split the words into three types, namely narrow query keywords (NK-words), narrow heading-words (NH-words), and heading keywords (HK-words). The NK-words are query keywords which are not heading words, and the NH-words are heading words which are not query keywords. The HK-words are the words which are heading words and also query keywords. We modify the baseline method explained before as described below.

Sentence Segmentation: Because headings and blocks are semantically coherent fragments and no sentence should overlap the boundaries of them, we segment the text contents of pages into text fragments by all their boundaries, and then segment the fragments into sentences by the Stanford CoreNLP toolkit [12]. Because we show headings in a different way from other components of snippets (as discussed later), we separately extract headings and other sentences.

Sentence Scoring: The new $\mathrm{score}(q, s)$ and $\mathrm{weight}(w, s)$ are calculated by:

$$\mathrm{score}(q, s) = \sum_{w \in q \cup h(s)} \frac{\mathrm{weight}(w, s)}{k_1 + \mathrm{weight}(w, s)} \log \frac{N - \mathrm{sf}(w) + 0.5}{\mathrm{sf}(w) + 0.5}, \tag{3}$$

$$\mathrm{weight}(w, s) = \frac{\mathrm{occurs}(w, s) \cdot boost^{\mathrm{typeof}(w)}}{\left((1 - b) + b \cdot \frac{\mathrm{length}(s)}{\mathrm{avgLength}}\right)} \tag{4}$$

where $h(s)$ is heading words of s, w is a word in q or $h(s)$, and $\mathrm{typeof}(w) \in \{\text{NH-words}, \text{NK-words}, \text{HK-words}\}$ is the type of w. The parameter $boost^{\mathrm{typeof}(w)}$ represents the importance of the occurrences of the words whose type is $\mathrm{typeof}(w)$.

Output: The generated text snippets and their headings including the title or URL of the input page. In case that heading structure of documents are given, we can improve readability of snippets by showing sentences and their headings separately [13]. We also adopt this idea. Figure 4 shows an example output.

The other steps of this method are same as those of the baseline method. We call this method the *existing* method.

4.3 Keyword Occurrences in Headings

Our observation is that the heading words are very often omitted from sentences. Despite such omission, heading words are important to clarify the topic of their associated sentences. Therefore, to select sentences that well represent the original document considering such omission, we must count keyword occurrences in the contextual headings of the sentences as well as in the sentences themselves.
Sentence Scoring: Based on this idea, we regard that each sentence comprises two fields, namely the contents of the sentence itself and its contextual headings, and adopt a variant of BM25F, a scoring function for documents comprising multiple fields [16]. The function calculates the score of a sentence S comprising two fields for keyword query q by the following formulas:

$$\mathrm{score}(q, S) = \sum_{\kappa \in q} \frac{\mathrm{weight}(\kappa, S)}{k_1 + \mathrm{weight}(\kappa, S)} \log \frac{N - \mathrm{sf}(\kappa) + 0.5}{\mathrm{sf}(\kappa) + 0.5}, \tag{5}$$

$$\mathrm{weight}(\kappa, S) = \sum_{f \in S} \frac{\mathrm{occurs}(\kappa, f, S) \cdot \mathit{boost}_f}{\left((1-b) + b \cdot \frac{\mathrm{length}(f,S)}{\mathrm{avgLength}(f)}\right)} \tag{6}$$

where f is a field in S, $\mathrm{occurs}(\kappa, f, S)$ is the number of occurrences of κ in f of S, boost_f is the weight of keyword occurrences in f, $\mathrm{length}(f, S)$ is the length of f in S, and $\mathrm{avgLength}(f)$ is the average length of f. The other steps of this method are same as those of the existing method. We call this method *our* method.

4.4 Combination of Two Advanced Methods

Above two modifications can be applied independently. Therefore, we can consider the fourth method which adopts both of them.
Sentence Scoring: We calculate the combined score and weight by:

$$\mathrm{score}(q, S) = \sum_{w \in q \cup h(S)} \frac{\mathrm{weight}(w, S)}{k_1 + \mathrm{weight}(w, S)} \log \frac{N - \mathrm{sf}(w) + 0.5}{\mathrm{sf}(w) + 0.5}, \tag{7}$$

$$\mathrm{weight}(w, S) = \sum_{f \in S} \frac{\mathrm{occurs}(w, f, S) \cdot \mathit{boost}_f^{typeof(w)}}{\left((1-b) + b \cdot \frac{\mathrm{length}(f,S)}{\mathrm{avgLength}(f)}\right)} \tag{8}$$

where $\mathit{boost}_f^{typeof(w)}$ is the weight of occurrences of w in f. The other steps are same as those of our method. We call this method the *combination* method.

4.5 Parameters and Fine Tuning

These scoring functions require three types of parameters: The saturation factor k_1 controls scaling of weighted term frequency, b controls the strength of length normalization, and *boost* controls the weights of term occurrences of each type of words in each field. Because the scaling and normalization are not the main topic of this paper, we use the default values 2.0 for k_1 and 0.75 for b [16].

Table 1. Boost for occurrence of words of each type in each field.

Parameter name	Value	Parameter name	Value
$boost_{\text{headings}}^{\text{HK-words}}$	3.0	$boost_{\text{sentence}}^{\text{HK-words}}$ ($boost^{\text{HK-words}}$)	4.0
$boost_{\text{headings}}^{\text{NH-words}}$	0	$boost_{\text{sentence}}^{\text{NH-words}}$ ($boost^{\text{NH-words}}$)	1.0
$boost_{\text{headings}}^{\text{NK-words}}$ ($boost_{\text{headings}}$)	3.0	$boost_{\text{sentence}}^{\text{NK-words}}$ ($boost^{\text{NK-words}}$, $boost_{\text{sentence}}$)	3.0

The setting of *boost* is important for effective heading-aware snippet generation. According to the observation by Pembe and Güngör [13], occurrences of query keywords are three times more important than those of heading words. Therefore, we use 3.0 for all $boost_{\text{sentence}}^{\text{NK-words}}$ in Section 4.4, $boost_{\text{sentence}}$ in Section 4.3, and $boost^{\text{NK-words}}$ in Section 4.2 while we use 1.0 for all $boost_{\text{sentence}}^{\text{NH-words}}$ in Section 4.4 and $boost^{\text{NH}-\text{words}}$ in Section 4.2. Because there is no existing observation about the balance of weights of the keyword occurrences in sentences and in their contextual headings, we simply use 3.0 (same as $boost_{\text{sentence}}^{\text{NK-words}}$) for $boost_{\text{headings}}^{\text{NK-words}}$ in Section 4.4 and $boost_{\text{headings}}$ in Section 4.3. Because heading words always occur in headings, we use 0 for $boost_{\text{headings}}^{\text{NH-words}}$. As the weight of HK-words, we use the summations of the weight of NH-words and the weight of NK-words. All the *boost* values are listed in Table 1 for reference.

5 Evaluation

In this section, we evaluate each snippet generation method.

5.1 Evaluation Methodology

As discussed in Section 4.1, judgeability is the most important property of effective search result snippets. Therefore, to measure the effectiveness of snippet generation methods, we measure the judgeability of their output snippets. To measure the judgeability, in the INEX snippet retrieval track [18], the results of relevance judgments under two different conditions are compared. One judgment is performed based on the entire documents while the other is only based on their snippets. If they agree, the snippets provided high judgeability and the snippet generation method was effective. We use this measure and also their length limit of snippets, which is 180 letters for a page.

However, the target of INEX is XML documents while our target is web pages. Therefore, we used a data set for text retrieval conference (TREC) 2014 web track ad-hoc task [6].

5.2 Data Set and Evaluation Measures

Queries and intents: Fifty keyword queries and their intent descriptions.
Document collection: ClueWeb12 B13, a web snapshot crawled in 2012. We extracted top-20 pages for each query (total 1,000 pages) from the official baseline

Table 2. Comparison of average evaluation scores of four methods.

Method	Recall	NR	GM
Baseline	**.475**	**.828**	**.512**
Exist.	.373	.780	.386
Ours	.438	.777	.456
Combi.	.396	.776	.401

Table 3. Average evaluation scores of four methods for each type of queries.

(A) For 24 *faceted* queries.

Method	Recall	NR	GM
Baseline	**.524**	**.806**	**.539**
Exist.	.416	.737	.378
Ours	.509	.723	.470
Combi.	.290	.737	.257

(B) For 24 *single* queries.

Method	Recall	NR	GM
Baseline	.431	**.837**	.488
Exist.	.336	.804	.392
Ours	.375	.816	.443
Combi.	**.491**	.795	**.530**

search result for the TREC task. The result is generated by the default scoring by Indri search engine and filtered by Waterloo spam filter.

Page-based relevance judgment data: The TREC official graded relevance of the entire pages to the intents. We simply regarded that documents whose grades are more than 0 as relevant to the intent, and the others are irrelevant.

Snippet-based relevance judgment data: We carried out a user experiment with four participants. They are all non-native English readers familiar with web search. In each period of the experiment, each participant is required to read the intent description behind a query first. Next, he is required to scan top-20 search result items containing the snippets generated by a method and to judge whether each original page is relevant to the intent. We broke out the search results to participants by Graeco-Latin square, therefore each snippet was not judged more than once, and each participant did not judge a page more than once and used all methods almost evenly. As described above, we adopted binary relevance. It is because the user of a real web search engine must decide to read or not for each original page based on its snippets and there is no intermediate choice.

Evaluation measures: We use three evaluation measures from the INEX track: Recall, negative recall (NR), and the geometric mean (GM) of them. Recall is the ratio of pages correctly judged as relevant on their snippets to pages relevant as a whole. It is calculated by $|\textit{Correctly judged pages relevant as a whole}|/|\textit{Pages relevant as a whole}|$. On the other hand, NR is the ratio of pages correctly judged as irrelevant on their snippets to pages irrelevant as a whole. It is calculated by $|\textit{Correctly judged pages irrelevant as a whole}|/|\textit{Pages irrelevant as a whole}|$. GM is the primary evaluation measure of the INEX track and our evaluation. It is calculated by $\sqrt{\textit{Recall} \cdot \textit{NR}}$. To integrate the evaluation scores for multiple queries, we calculated the arithmetic mean of them.

5.3 Evaluation Results and Discussion

Comparison of snippet generation methods: First, we compared the average evaluation scores of four snippet generation methods. Table 2 lists the results. The baseline method achieved the top scores by all the evaluation measures. Our heading-aware method achieved the second GM score. The existing heading-aware method achieved the worst GM score and its difference from the baseline method was statistically significant ($p < 0.05$) according to Student's paired t-test where each pair is composed of the evaluation scores of the baseline

Table 4. Average GM scores of four methods and query length excluding stopwords.

$\|Keywords\|$	$\|Queries\|$	Baseline	Exist.	Ours	Combi.
2	25	**.585**	.406	.543	.393
3	10	**.503**	.387	.378	.388
4 or more	15	.394	.350	.362	**.425**

and heading-aware methods for a query. Hereafter in this paper, we discuss statistical significance based on the same test procedure. There was no statistically significant difference from the baseline to the other methods. As shown in this result, the heading-aware methods were not effective for general queries. The difference of the GM scores was mainly caused by the difference of the recall scores. In fact, the best method improved the recall score by 27.3% from the worst while the NR score by only 6.70%. In other words, the effectiveness of the methods mainly depends on how many relevant pages its output snippets can indicate to the users. This tendency was seen through all evaluations.

Effect of query type: For detailed evaluation, TREC splits queries into several types. *Faceted* queries are underspecified, while there are clear and focused intents behind *single* queries [6]. The data set contains 24 queries of each type. It also contains only two *ambiguous* queries, however we ignored them. The scores for the faceted queries are listed in Table 3 (A) and the scores for the single queries are listed in (B). As shown in these tables, the baseline method achieved the best scores for faceted queries while the combination method achieved the best recall and GM scores for single queries. Only the GM score difference between the baseline and combination methods for faceted queries was statistically significant. This fact suggests that heading-aware snippet generation methods may be effective for clearly specified intents. To indicate the relevance of a page to a clearly specified intent, small number of sentences and their rich contextual information, i.e. their headings, may be important. In the other cases, it may be important to show a larger number of sentences in the page. For further discussion, another evaluation with more queries is needed.

Effect of query length: When a user inputs multiple keywords, the user is probably requesting pages in which all the keywords occur in relation to each other. On the other hand, as discussed in Section 4.3, contextual heading words have semantic relationship to their associated sentences. Therefore, the heading-aware methods must be more useful for queries containing more keywords. In other words, there are usually less sentences containing more different keywords directly. However, considering the contextual headings of the sentences, heading-aware methods can detect more of relevant sentences. Based on this idea, we classified the queries by their numbers of keywords excluding stopwords. Table 4 lists the numbers of queries in each class and the GM scores of each method for each class. For the queries with two keywords, the baseline method achieved the best GM score and its differences from the existing and combination methods were statistically significant. However, only the combination method retained its score for the longer queries while the other three methods lost their scores.

Table 5. Median amount of required time in second to check snippets of 20 pages for one query.

(A) By each method.

Method	Time in sec.
Baseline	411.5
Exist.	**308.5**
Ours	315.5
Combi.	349.0

(B) By each participant.

Participant	Time in sec.
A	297.5
B	429.0
C	347.5
D	293.0

Table 6. Comparison of average evaluation scores of four participants.

Participant	Recall	NR	GM
A	.367	.787	.376
B	.482	.783	.498
C	.459	.815	.451
D	.375	.776	.430

The correlation coefficient of the GM score and the number of pairs of different query keywords for each query was .247 for the combination method while -0.105 for the baseline. Especially, for four or more keywords, the combination method achieved the best score. It supports the above discussion about longer queries. For further discussion, another evaluation with more queries is needed.

Query type and query length: Query type depends on query length because more query keywords specify the intents of the query more clearly. In fact, the average length of single queries was 3.54 words while that of faceted queries was 2.25 words. Note that this dependence might affect our evaluation results.

Required time analysis: We also measured the median required time for checking 20 pages for a query. Table 5 (A) lists the results. Intuitively, the assessors took much more time for our evaluation tasks than practical search tasks. It may be because they are non-native English reader, and/or because they read snippets more carefully for more accurate judgment than usual. Generally, heading-aware snippets significantly reduced the required time. It must be because the users can read structured text more easily than plain text.

Effect of assessors: We also compared the required time and evaluation scores for each assessor. Table 5 (B) lists the median time in second required for checking 20 pages and Table 6 lists the average evaluation scores for each assessor. As shown in Table 5 (B), the required times are quite different for each assessor. The difference also affects the average GM scores of them, that is, the most and second-most careful assessors, B and C, achieved the best and second-best GM scores respectively. Note that the effect of the assessors for the other comparative evaluations is limited because each assessor uses each methods almost evenly.

6 Conclusion

We introduced a novel idea for heading-aware snippet generation and compared one baseline and three heading-aware snippet generation methods. The idea is that sentences whose contextual headings contain query keywords provide judgeability as well as sentences containing query keywords directly. Our evaluation result indicated that the heading-aware methods were not effective for general queries. Only for queries representing its intents clearly or containing four or more keywords, the heading-aware combination method achieved the best score. This fact suggests that heading-aware snippet generation is useful

for such queries. However, to discuss the statistical significance of the result, an additional evaluation with more queries is needed.

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

References

1. Ageev, M., Lagun, D., Agichtein, E.: Towards task-based snippet evaluation: Preliminary results and challenges. In: MUBE (SIGIR Workshop). pp. 1–2 (2013)
2. Amer-Yahia, S., Lalmas, M.: XML search: Languages, INEX and scoring. SIGMOD Rec. 35(4), 16–23 (2006)
3. Arvola, P., Kekäläinen, J., Junkkari, M.: Contextualization models for XML retrieval. Inf. Process. Manage. 47(5), 762–776 (2011)
4. Carbonell, J., Goldstein, J.: The use of MMR, diversity-based reranking for reordering documents and producing summaries. In: SIGIR. pp. 335–336 (1998)
5. Clarke, C.L.A., Agichtein, E., Dumais, S., White, R.W.: The influence of caption features on clickthrough patterns in web search. In: SIGIR. pp. 135–142 (2007)
6. Collins-Thompson, K., Macdonald, C., Bennett, P.N., Diaz, F., Voorhees, E.M.: TREC 2014 web track overview. In: TREC (2014)
7. Kanungo, T., Orr, D.: Predicting the readability of short web summaries. In: WSDM. pp. 202–211 (2009)
8. Leal, L., Scholer, F., Thom, J.: RMIT at INEX 2011 snippet retrieval track. In: INEX. pp. 300–305 (2011)
9. Liang, S., Devlin, S., Tait, J.: Evaluating web search result summaries. In: Adv. in Info. Retrv., pp. 96–106. Springer (2006)
10. Lu, W., Robertson, S., MacFarlane, A.: Field-weighted XML retrieval based on BM25. In: INEX. pp. 161–171 (2006)
11. Manabe, T., Tajima, K.: Extracting logical hierarchical structure of HTML documents based on headings. VLDB 8(12), 1606–1617 (2015)
12. 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)
13. Pembe, F.C., Güngör, T.: Structure-preserving and query-biased document summarisation for web searching. Online Info. Rev. 33(4), 696–719 (2009)
14. Porter, M.F.: An algorithm for suffix stripping. In: Readings in Info. Retrv., pp. 313–316. Morgan Kaufmann Publishers (1997)
15. Robertson, S., Walker, S., Jones, S., Hancock-Beaulieu, M., Gatford, M.: Okapi at TREC-3. In: TREC. pp. 109–126 (1996)
16. Robertson, S., Zaragoza, H., Taylor, M.: Simple BM25 extension to multiple weighted fields. In: CIKM. pp. 42–49 (2004)
17. Tombros, A., Sanderson, M.: Advantages of query biased summaries in information retrieval. In: SIGIR. pp. 2–10 (1998)
18. Trappett, M., Geva, S., Trotman, A., Scholer, F., Sanderson, M.: Overview of the INEX 2013 snippet retrieval track. In: CLEF (2013)
19. Wang, S., Hong, Y., Yang, J.: Pku at inex 2011 xml snippet track. In: INEX. pp. 331–336 (2011)
20. Zhang, L., Zhang, Y., Chen, Y.: Summarizing highly structured documents for effective search interaction. In: SIGIR. pp. 145–154 (2012)