Concept-based Medical Document Retrieval: THCIB at CLEF eHealth Lab 2013 Task 3

Xiaoshi Zhong[1], Yunqing Xia[1], Zhongda Xie[1],
Sen Na[2], Qinan Hu[2] and Yaohai Huang[2]

[1] Dept. of Comp. Sci. & Tech., Tsinghua National Laboratary of Information Science and Technologies, Tsinghua University, Beijing 100084, China
{xszhong, yqxia, zdxie}@tsinghua.edu.cn

[2] Canon Information Technology (Beijing) Co. Ltd., Beijing 100080, China
{nasen, huqinan, huangyaohai}@canon-ib.com.cn

Abstract. We describe our participation in the task 3 of ShARe/CLEF eHealth Lab 2013: information retrieval to address questions patient may have when reading clinical reports. In our experiments, we focus mainly on two levels of analysis, namely query analysis and document analysis, to disclose the relevance between query and documents. In terms of query analysis, we first observe each medical-oriented query to find its identical or related UMLS concepts derived from the query, which may help to induce relevant results that refer to the same thing but are represented in other surface forms. In such manner, we extend the query based on the medical concepts so as to achieve a bigger coverage. In terms of document analysis, we leverage different scores (e.g., relevance score, PageRank score, HITS score and layout score) as feature to re-rank the documents of search results. With those two levels of analysis, we implement a concept-based method and a topic-based method to accomplish the task of medical document retrieval. Experiments indicate that the proposed method is effective.

Keywords: medical document retrieval, concept, query analysis, topic, document ranking

1 Introduction

Research on medical document retrieval becomes increasingly hot nowadays. Patients are always curious of what is exactly written on the discharge summaries and what the medical text exactly means. A headache issue is that the medical text is very professional and hard to follow. As an effective manner in answering the questions from the patients, medical information retrieval becomes highly popular.

The THCIB team, comprised of researchers from Intelligent Search group at Tsinghua University and Canon Information Technology (Beijing) Co. Ltd. participated in all the three tasks in ShARe/CLEF eHealth Lab 2013 [1]. In this technical report, we describe our solution to task 3, namely retrieving information to address the questions that patient may come up with when they read the medical

reports. The goal of this task is to produce a ranking list of documents with respect to the most relevant aspects of a given medical query. We submitted seven runs for this task. One is baseline, which uses only query and description, conducts no query expansion, and ranks documents according to relevance and PageRank. The remaining six runs are different implementations of our system, in which three runs use discharge summaries and other three ones do not. In our system, we conduct concept-based query analysis and query expansion. We rank the medical documents with a unified ranking model considering relevance score, PageRank score, HITS score and layout score. We also conduct two runs based on a topic-based query expansion and rank the documents according to relevance.

Based on query analysis, we developed two medical document retrieval methods: concept-based method and topic-based method. In the concept-based method, we first attempt to extend each query by extracting the UMLS[1] concepts it belongs to and the synonyms of these concepts so as to induce an expanded search query. After searching documents with the expanded query, we re-rank the search results by a unified ranking model, which is a linear combination of several feature scores: relevance score, PageRank [2] score, HITS [3] score and layout score (derived from HTML structure [4]). In the topic-based method, we conduct query analysis with the manually defined seven topics, each of which is represented by several topic words. With these topics, we extend each query to a set of expanded queries, which are used to search medical documents.

In the remainder of this paper, Section 2 demonstrates our infrastructure for pre-processing the medical documents. Section 3 describes our method for the task and Section 4 summarizes our submitted runs. We present the evaluation and discuss the possible problem in Section 5. Finally, we make conclusions and state the future work in Section 6.

2 Text Preprocessing

Our participation in the CLEF eHealth 2013 task 3 builds upon the document collection provided by the organizer. We use HTMLParser[2] to extract the content and URLs from the HTML pages. The plain text is used for document indexing with Lucene[3], while URL links are used in the PageRank and HITS algorithms.

In the process of document indexing, we only extract two kinds of HTML text: title and the plain text of HTML page. This is a straightforward solution as it is difficult to seek for a unified framework with several HTML fields to describe all the webpages. We leave it as our future work to find a better way to efficiently model the HTML pages [5][6]. In addition, in order to record the information of an HTML page such as UID and URL, we finally index four fields of the original HTML pages: ***title***, ***content***, ***UID***, and ***URL***.

[1] UMLS: Unified Medical Language System, http://www.nlm.nih.gov/research/umls/
[2] http://htmlparser.sourceforge.net/
[3] http://lucene.apache.org/

3 Methodology

For the specific task of retrieving a ranked list of documents response to a medical query with description information such as scenario and narrative, as shown in Fig.1, we implement two different methods for the goal of medical document retrieval, as mentioned in Section 1: 1) extracting concepts of a query and synonyms of those concepts to extend a query and 2) manually defining topics to extend a query.

```
<query>
    <id>qtest12</id>
    <discharge_summary>07726-023607-
        DISCHARGE_SUMMARY.txt</discharge_summary>
    <desc>is clots in jugular in connection with HIV</desc>
    <narr>clots in jugular and HIV</narr>
    <profile>The 46-year old woman with end stage renal disease and polysubstance
        abuse is depressed and wants to find out about her chance of having HIV.
    </profile>
</query>
```

Fig.1. A sample medical query.

The main idea of the two methods lies in two aspects: query analysis and document analysis. In terms of analysis, we observe a query according to its description information (e.g., text of <desc> field in Fig.1) and other resources (e.g., UMLS) to expand the query so as to obtain more relevant queries. By doing this, we achieve two goals. The first one is to induce more relevant queries that are either specific ones to the original query or different ones but expressing the same meaning; the second one is to leverage extra information to explicitly redefine the query. For the document analysis, we derive several features from document as scores to evaluate and re-rank the search results. In what follows, we explain our method in details.

3.1 Concept-based Query Expansion for Document Retrieval

The organizer provides medical-oriented queries extracted from discharge summaries (e.g., <discharge_summary> field in Fig.1) with some description information, as shown in Fig.1. We believe it is necessary to make full use of the information to improve performance. Similar to Wang et al. (2013) [7], our method extracts concepts of a query and induces synonyms of those concepts from UMLS. After that, we integrate those concepts and synonyms into an expanded search query. For example, for the query in Fig.1, we can extract two concepts, i.e., “clots” and “HIV”, from UMLS. There are three synonyms of the concept “clots”, i.e., “Clotrimazole”, “1H-Imidazole” and “Klotrimazole”, and forty-five synonyms of the concept “HIV”, including “Positivity, HIV Antibody”, “HTLV-III Seropositivities” and so on. According to the <desc> or <narr> or both, we select some of above concepts and their synonyms, noted as ***expansion terms***.

After selecting concepts and synonyms, we integrate the ***expansion terms*** to the original query to get a new search query. The integrating strategy is straightforward.

That is only simply split-joint all the ***expansion terms*** and original query, shown as Algorithm 1.

Algorithm 1:
Set SQ = *query*
For all the expansion terms $k \in [1, K]$, do
$SQ = SQ$ # "ET_k"

In the algorithm, SQ represents search query, K denotes the number of ***expansion terms***, and ET_k represents the k^{th} expansion terms. The symbol "#" here represents space character (i.e., 0x20), and the double quotation marks indicate that the string in it must appear consecutively.

As we see that, "HIV" actually is short for "*Human immunodeficiency virus*" and among all the queries there are only seven those acronyms, we choose to recognize and normalize the abbreviations/acronyms based on the tool that we developed for CLEF 2013 eHealth Task 2 (i.e., normalization of abbreviations / acronyms) [8]. We extended this tool by incorporating online resources such as Google[4] and Wikipedia[5] in abbreviation / acronym normalization. For those recognized abbreviations / acronyms, if they cannot be normalized by UMLS. We searched them in Google and Wikipedia. After simple text analysis, we extract the full version of the abbreviations/acronyms. Table 1 gives the acronym/full term mappings for the official queries. To be specific, if the original query contains an abbreviation/acronym, we add its full term to the search query, i.e., $SQ = SQ$ # "full term".

Table 1. Acronym / full term mappings.

Acronym	Full term
HIV	Human immunodeficiency virus
SOB	Shortness of breath
ASA	Aspirin
MI	Myocardial infarction
COPD	Chronic obstructive pulmonary disease
Hypo-	Hypo-glycemia
HA	Headache

Let's take the query in Fig.1 as example. For illustration convenience, we assume that the ***expansion terms*** comprised only five terms: "clots", "HIV", "Clotrimazole", "Positivity, HIV Antibody", "HTLV-III Seropositivities", as well as the full term "*Human immunodeficiency virus*" of "HIV". After applying the integrating strategy to those expansion terms, we obtain the ***search query*** as follows:

clots in jugular and HIV # "clots" # "HIV" # "Clotrimazole" # "Positivity, HIV Antibody" # "HTLV-III Seropositivities" # "Human immunodeficiency virus",

which is denoted by SQ_{hiv}.

According to the analysis of Section 2, we extract two kinds of HTML information (i.e., title and content) and resort to Lucene, to index the two fields (i.e., ***title*** and

[4] http://www.google.com
[5] http://en.wikipedia.org

content) using the multi-field search function for retrieving. The ***search query*** (i.e., SQ_{hiv}) we obtain is used for searching in the "content" field.

In general, a ***search query*** (e.g., SQ_{hiv}) is likely to be more generalized to describe concept(s) from different perspective, so does the function of content of a document. It is widely accepted that a term, especially a concept, occurring in the title of a document contributes much more than the one occurring in the content. Therefore, besides the generalized searching ***content*** field with ***search query*** (e.g., SQ_{hiv}), we also search ***title*** field in an exact matching manner with concepts derived from the original query. Taking the example above again, we apply the multi-field search method to the two fields of ***title*** and ***content*** by searching the ***search query*** (i.e., SQ_{hiv}) and ***concepts*** (i.e., "clots" + "HIV"). With this search strategy, we finally obtain a ranked list of medical documents, denoted as ***search results***.

Up to now the analysis in this subsection is about the query analysis. In what follows, we move forward to the level of document analysis, whose goal is to re-rank the ***search results***.

In case of document analysis, we derive and calculate four kinds of feature scores to re-rank the search results, namely, Lucene ***relevance feature*** , ***global*** PageRank ***feature***, ***local*** HITS ***feature*** and HTML ***layout feature***. Due to those popular terms like Lucene, PageRank algorithm [2], and HITS algorithm [3], we think it is just necessary to explain what the "HTML ***layout feature***" means.

Layout Feature (HTML). It assumes that the term occurring either in title or subtitle, or being highlighted or set bold or italic in an HTML page should be more important and receives a greater weight than other common terms if using term weight to represent the HTML[4]. Table 2 shows the signal fields of HTML. We take them into account in ***layout feature*** calculation.

Table 2. Feature field in an HTML page.

Feature field	Description
Title	Text between tag <title> and </title>
Subtitle	Text between tag and
Anchor text	Text between tag <a> and
Italic	Text between tag and
Underline	Text between tag <u> and </u>
Bold	Text between tag and

Even within those fields shown in Table 2, a term in different field should be assigned a different weight. Usually, it is experiment oriented and demands to tune with gold data. But there is no suitable gold data matching the first year of the task 3 in CLEF eHealth 2013.Thus, we simply assign value 1 to a term of the ***expansion terms*** if the term occurs in one of those six fields, and the value is capable of accumulation. For example, if the term of expansion terms "HIV" occurs in Title, Subtitle, Bold and Underline fields, then the term "HIV" will get a layout score of 4. Sum all the layout scores of all the expansion terms to get the layout score of a document. Note the layout score of document d as ***Layout***(d).

Searching Feature (Lucene). We obtain the search results with relevance score by Lucene4.3 from the query analysis phase. We use both the BM25 and VSM similarity based on *tf-idf* statistics during searching. Note the BM25 score and VSM score of document d as ***BM25***(d) and ***VSM***(d), respectively.

Global Feature (PageRank). We apply the PageRank algorithm [2] to all the collection of documents to get a global weight of all the documents. Note the *PageRank* score of document *d* as ***PageRank***(*d*).

Local Feature (HITS). We apply the HITS algorithm [3] to the set of documents from ***search results*** and implemented without any URL expansion on the Internet. With the HITS algorithm, we produce two scores of a document *d*, namely Hub and Authority, denoted by ***Hub***(*d*) and ***Authority***(*d*), respectively.

The motivation of combining ***global features*** and ***local features*** lies in that we assume that there are quite a large number of Web pages on the Web, and a Web page could be evaluated by the other Web page on the Web or in a limited scope. The PageRank algorithm may hold advantage in the global collection while HITS algorithm is appropriate for the local collection, to some extent.

As different feature scores hold different ranges, it is necessary to normalize them into a unified format so as to compare them with each other. Let $OS(d_i)$ denote an original score of the i^{th} document d_i, our normalization algorithm is given as follow:

$$uOS(d_i) = \frac{OS(d_i)}{\sqrt{\sum_{i=1}^{N} OS(d_i)}} \quad (1)$$

$$maxOS = max\{uOS(d_i)\} \quad (2)$$

$$nOS(d_i) = mOS(d_i)/maxOS \quad (3)$$

where $uOS(d_i)$ denotes the i^{th} element of the unit vector, *maxOS* represents the element with maximum value within the unit vector, $nOS(d_i)$ denotes the value after our normalization strategy. We first normalize the original score into unit vector format. Then we divide each element of the unit vector by the element with maximum value. With this normalization strategy, $nOS(d_i) \in (0,1]$, $max\{nOS(d_i)\}$ is equal to 1 and score from different format can be computed together. In this way, we can produce the normalized score of the feature scores, denote them as *nLayout*(*d*) *nBM25*(*d*), *nVSM*(*d*), *nPageRank*(*d*), *nHub*(*d*) and *nAuthority*(*d*), respectively.

After normalizing all the feature scores into a unified format, we re-rank the documents of search results by a method of linear score combination. In our experiment, the strategy of score combination is as Eq(4),

$$score(d_i) = \alpha * 0.5 * (nBM25(d_i) + nVSM(d_i) + \beta * nPageRank(d_i) + \gamma * 0.5 * (nHub(d_i) + nAuthority(d_i)) \quad (4)$$

where α, β, γ are the coefficients, which are required to be tuned with the assistance of a gold data, while we simply set the value 1 to all of them due to a lack of a gold data as mentioned above.

3.2 Topic-based Query Expansion for Document Retrieval

Taking a closer look at the given queries and their corresponding description information, such as a sample query in Fig.1, as well as the discharge summary file, we find that such a query can be categorized into one of seven topics according to description information. The topics as well as the associated topic words are presented in Table 3, which are compiled manually. Note that the compilation is carried out independently. We compiled the topics and keywords based on study on medical Web data.

Following the idea in [9], we simply classify a query to one of the above topics as follows.

1) Cumulate the times of the topic words of each topic occurring in the <desc> field of a query, noted as N_i (i=1,…, 7), where *i* denotes the topic ID.

2) Select the maximum value of N_i, noted as $max\{ N_i \}$.

3) If $N_k = N_j = max\{N_i\}$ $(k \neq j)$, select the most common topic which contains more topic words.

4) If max{Ni} = 0, select the topic which contains most topic words.

Table 3. Topics and topic definitions.

ID	Topic	Topic words
1	What	what is, learn about, disease type, type of disease, medical term, definition, about, overview, learn more about, patient information, introduction, information for patients
2	Reason	common causes, symptoms, symptom, causes, diagnosis, cause, signs, what causes, diagnosed, disorders that cause, etiology
3	Pain	effects, effect
4	connect	connection, connect, connections
5	treatment	How to treat, treatments, treatment, drugs, medications, treated
6	Avoid	at risk for, avoid, prevent, prevention, precaution, prophylaxis
7	Care	Risk factors, complications, potential dangers, possible complications

For instance, only topic word "connection" of topic "connect" occurs in the text of <desc> shown in Fig.1, the $max\{N_i\}$ is 1 where $i = 4$, therefore the corresponding query "*clots in jugular and HIV*" belongs to the topic "connect".

After assigning a topic to the query, we leverage concepts of the query and those topic words of the assigned topic to extend the query. The query expansion strategy is also simple, integrating each topic word a time and all the concepts to get a ***search query***, for all the topic words, we get a set of search queries as shown in Algorithm 2.

Algorithm 2:

```
for all topic words k ∈ [1, K] do
  Set SQ = TWk
for all concepts n ∈ [1, N] do
SQ = "SQ" + "CCn"
```

where K means the number of topic words of the assigned topic, SQ means search query, TW_k represents the k^{th} topic word, N means the number of concepts derived from the original query, and CC_n represents the n^{th} concept. For example, in the case of query "*clots in jugular and HIV*", which has two concepts "clots" and "HIV", based on the integrating strategy, we can produce a search query like "connection" + "clots" + "HIV" by combining the first topic word of the assigned topic and the concepts of the query. For this example, we finally induce three search queries due to three topic words of the topic "connect". The double quotation marks indicate that the string must appear consecutively. And the plus sign (+) suggests that the strings must be co-occurring.

Obtaining a list of documents with relevance scores (BM25 of Lucene4.2) with respect to a search query, we get three such document lists of query "*clots in jugular and HIV*" in total. As some documents may appear in different lists, we simply sum all the relevance scores of a same document in different lists together to merge those documents into a list and re-rank the documents. Take the above query as example again, assuming that relevance score of document d_i in the list j is S_{ij}, the final score of document d_i is calculated as Eq(5):

$$score(d_i) = \sum_{j=1}^{N} S_{ij} \quad \textbf{(5)}$$

In the example of above query, the value of N is 3. With finally scoring the documents, we can obtain a final list of document result response to a given query.

4 Submitted Runs

In total, we submitted seven runs to the task 3 of CLEF eHealth 2013:

1) Run 1: Baseline method. We only use the query to implement ad hoc retrieving and combine the scores (i.e., relevance score, PageRank score and HITS score) with a linear method in document ranking.
2) Run 2: Concept-based method. We use the description (<desc>, see in Fig.1) as well as discharge summaries (<discharge_summary>) to analyze each query into expanded queries and re-rank the search results by combining the scores (i.e., relevance score, layout Score, PageRank score and HITS score) with a linear method. UMLS concepts play a vital role in query analysis and expansion.
3) Run 3: Concept-based method. We further use narrative (<narr>) to analyze each query into expanded queries. The re-ranking method is same to Run 2.
4) Run 4: Topic-based method. Use description (<desc>) and narrative (<narr>) as well as discharge summaries (<discharge_summary>) to determine which topic the query belongs to. Documents are ranked only with relevance.
5) Run 5: Concept-based method. We exclude discharge summaries (<discharge_summary>) from Run 2.
6) Run 6: Concept-based method. We exclude discharge summaries (<discharge_summary>) from Run 3.

7) Run 7: Topic-based method. We exclude discharge summary (<discharge_summary>) to Run 4.

5 Evaluation

5.1 Dataset

The dataset for task 3 consists of a set of medical-related Web documents, provided by the Khresmoi project[6]. This collection covers a broad set of medical topics, but contains no patient information. The documents in the collection are downloaded from several online sources, including Health on the Net organization certified websites, the well-known medical sites and databases (e.g. Genetics Home Reference, Clinical.gov, Diagnosia)[7].

5.2 Metric

The official evaluate metric used in task 3 is p@N, which is a traditional metric in information retrieval. p@N indicates the percentage of relevant documents within the top *N* results. In our experiments, N is assigned 5, 10, 15, 20 and 30, respectively.

5.3 Official Results and Discussions

System performance
Fig.2 presents the official evaluation results of our system in seven runs for the task.

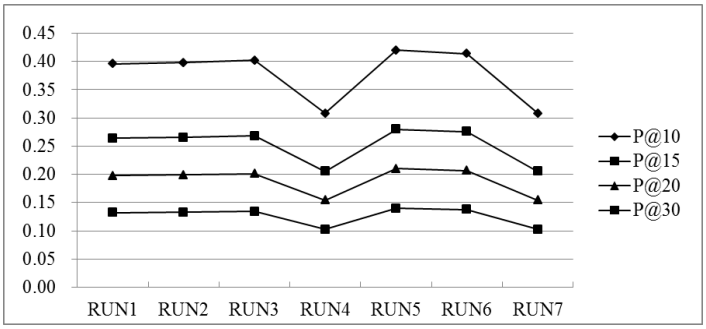

Fig.2. Evaluation results of THCIB medical document retrieval system.

Three observations are conducted as follows. First of all, we find in Fig.2 that performance of all runs uniformly drops when N bcomes bigger. This indicates that in

[6] http://www.khresmoi.eu/
[7] https://sites.google.com/site/shareclefehealth/data

medical document retrieval, bigger N does not bring more gain. It thus makes common sense to our knowledge that when the top number of searching documents is increasingly larger, the impact of features we used will be weaker and the random factor will be added.

Secondly, we observe on performance of our system with the concept-based method (i.e., Run 2, 3, 5 and 6). Seen from Fig.2 that the concept-based method outperforms the baseline (i.e., Run 1) slightly. Run 5 improves baseline most with 0.024. This confirms with us that concept indeed makes some contribution to information retrieval. We can also find in Fig.2 that Runs using query description and discharge summaries (i.e., Run 5 and 6) outperform ones not using (i.e., Run 2 and 3). This indicates that query description and discharge summaries bring noise in query expansion, which causes performan loss.

Thirdly, we observe on performance of our system with the topic-based method (i.e., Run 4 and 7). Fig.2 indicates that the topic-based method causes a significant drop (i.e., 0.086) compared to the basline. This is because the gold answers are generated with the the pool of results in Run 1, 2 and 5 of all systems. This is obviously unfair to Run 4 and Run 7. When we look into the evaluation results in Run 4 and 7, we surprisingly find that amongst those pooled, 154 results are judged relevant and 154 irrelvant in Run 4. In Run 7, irrelevant results are even fewer. Details are given in Table 4.

Table 4. Results details containing evaluation on relevant, irrelevant and un-judged.

Type	RUN1	RUN2	RUN3	RUN4	RUN5	RUN6	RUN7
Relevant	198	199	201	154	210	207	154
Irrelevant	302	301	226	154	290	259	150
Un-judged	0	0	73	192	0	34	196
p@10	0.396	0.398	0.402	0.308	0.42	0.414	0.308

It is interesting if we take into account of the situation in which all search results are pooled and judged by human. For results returned by Run 4 and 7, they are unfortunately not considered as answer candidates and not judged by human. If this could happen, we would expect a performance value being close to or higher than the best value in Run 5 though the judgment depends on human assessments.

Per-topic analysis

We conduct the analysis with Run 5. Per-topic comparison between Run 5 and other systems is given in Fig.3, in which the height of a bar in Fig. 2 given by:

Grey bars: $height(q) = our\_p@10(q) - median\_p@10(q)$

White bars: $height(q) = best\_p@10(q) - median\_p@10(q)$.

We can see in Fig.3 that Run 5 has 24 queries perform better than the median, especially achieves the best performance on 5 queries (i.e., query 6, 21, 26, 39 and 44), while 14 queries perform worse than the median, and other 12 queries perform in the median line. It means that in the specific task, medical document retrieval, our concept-based method can do something but heavily needs improvement.

In terms of the best performance of each query, it is quite difference among different queries. For example, some queries can achieve a very high result like query 1, 12, 24 and 32 (more than 70%), while some queries perform only normal or just so-so, like query 4, 6, 8, 20 and 45 (only about 10%), but in case of query 48, we guess, all the submitted documents are not relevant to the query. It may indicate that there is much work to do about medical intelligence.

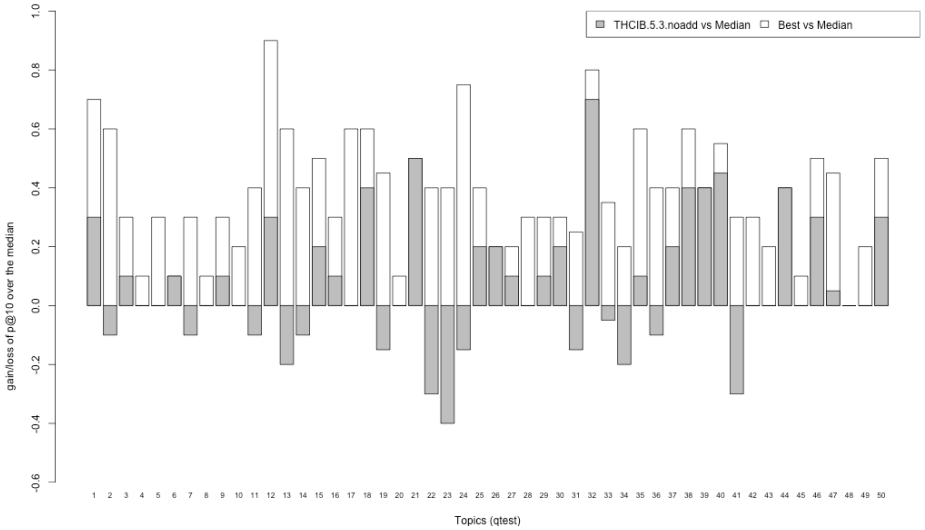

Fig.3. Per-topic comparison between Run 5 and the other systems.

6 Conclusion and future work

In this paper, we describe our medical document retrieval system for task 3 in CLEF eHealth 2013. Based upon query analysis and document analysis, we have developed two methods to implement our experiments, namely, concept-based method and topic-based method. Experimental results confirm our motivation. However, there is still some work that we can do in the future to improve the performance of medical document retrieval. For instantce, we will use BM25F to index the document considering the structure of HTML.

Acknowledgement

This research is supported by Canon Inc. (No. QIM2013). The Shared Annotated Resources (ShARe) project is funded by the United States National Institutes of

Health with grant number R01GM090187. We also appreciate the valuable comments from the task organizer.

References

1. Suominen, Hanna, Sanna Salanter, Sumithra Velupillai, Wendy W. Chapman, Guergana Savova, Noemie Elhadad, Danielle Mowery, Johannes Leveling, Lorraine Goeuriot, Liadh Kelly, David Martinez and Guido Zuccon. Overview of the ShARe/CLEF eHealth Evaluation Lab 2013. Proceedings of CLEF 2013. Lecture Notes in Computer Science (LNCS), Springer.
2. Brin, Sergey, and Lawrence Page. The anatomy of a large-scale hypertextual Web search engine. Computer networks and ISDN systems 30.1 (1998): 107-117.
3. Kleinberg, Jon M. Authoritative sources in a hyperlinked environment. Journal of the ACM (JACM) 46.5 (1999): 604-632.
4. Cutler, Michal, Yungming Shih, and Weiyi Meng. Using the structure of HTML documents to improve retrieval. In Proceedings of the USENIX symposium on internet technologies and systems. Vol. 12. 1997.
5. Robertson, Stephen, Hugo Zaragoza, and Michael Taylor. Simple BM25 extension to multiple weighted fields. In Proceedings of the thirteenth ACM international conference on Information and knowledge management. ACM, 2004.
6. Lu, Wei, Stephen Robertson, and Andrew MacFarlane. Field-weighted XML retrieval based on BM25. Advances in XML Information Retrieval and Evaluation. Springer Berlin Heidelberg, 2006. 161-171.
7. Junjun Wang, Guoyu Tang, Yunqing Xia, Qiang Zhou, Thomas Fang Zheng, Qinan Hu, Sen Na, Yaohai Huang. Understanding the Query: THCIB and THUIS at NTCIR-10 Intent Task. In Proceedings of NTCIR-10(2013).
8. Xia, Yunqing, Xiaoshi Zhong, Peng Liu, Cheng Tan, Sen Na, Qinan Hu, and Yaohai Huang . Normalization of Abbreviations/Acronyms: THCIB at CLEF eHealth 2013 Task 2. CLEF 2013 eHealth Lab working note.
9. Song, Wei, Yu Zhang, Handong Gao, Ting Liu, Sheng Li. HITSCIR System in NTCIR-9 Subtopic Mining Task. In Proceedings of *NTCIR-9*(2011).