Exploration of Efficient Vector Space Model in Medical IR

Na Gong
Mannheim Universität
B6, 68159 Mannheim
GERMANY
ngong@uni-mail

Qian Xia
Mannheim Universität
B6, 68159 Mannheim
GERMANY
qixia@uni-mai

Siying Liu
Mannheim Universität
B6, 68159 Mannheim
GERMANY
siliu@uni-mai

Abstract

Space vector model is an algebraic information retrieval model. It maintains the text filtering and ranking liability of general IR model meanwhile supports the development of many other advanced IR model. However, expensive time consumption is a main drawback of SVM in ranking area. The purpose of this project is to experiment three SVM speed-up approaches of tiered index, pre-clustering and random projection, which aims to explore their trade off between ranking performance and time consumption. In terms of the testing results based on a public medical text dataset NFCorpus, tiered index and pre-clustering are more competitive than random projection in VSM speed-up with less performance loss.

1 Introduction

Vector Space Model (VSM) is a basic algebraic information retrieval (IR) model wildly applied in information filtering and relevance ranking areas. A large amount of advance IR models such as latent semantic analysis, random indexing and Rocchio classification are developed based on VSM. Our project is to explore the different SVM speed-up approaches by testing their trade off between time consumption and ranking performance. All the models were implemented on a NFCorpus data set (English language) which contains in total 3,237 nutrition related queries with 134,295 multi-level relevance judgments for 5,371 medical documents (Boteva et al., 2016).

2 Data Structure

In total six experimental data sets were used to build the model and evaluate corresponding performance.

doc_dump. it contains all 5371 different medical documents with features of *id, url, title and abstract.*

nfdump. it involves in total 3437 nutrition related queries with feature of *id, url, title, maintext, comments, topics_tags, description, doctors_note, article_links, question_links, topic_links, video_links, medarticle_links.*

2-1-0.qrel *3. those three data sets are relevance judgment files for 3 levels divided by testing, training and development subsets.

Stopwords. a given stopword list is used for preprocessing.

3 Preprocessing

In order to reduce the data interference with construction of ranking models, seven steps of text preprocessing procedures were implemented. The first step was to clean data by removing queries without relevance judgments. At the end, the data set has 3237 queries. The next step is to select features of *doc_dump* and *nfdump* file based on gained domain knowledge. All medical documents data are from a same website hence all document instances have same URL address followed by different web id. By considering its highly repetitiveness and limited semantic information, the url feature was removed from *doc_dump*. Similarly, due to the highly feature similarity and large amount of missing value, *url, article_links, question_links, topic_links* are removed from *nfdump*. Particularly, *medical_links* of query file is exactly

the url feature of document file. To avoid its biased impact on similarity estimation between documents and queries, it was also excluded in query file. Afterwards, to keep in line with original preprocessing procedure, stopwords were moved out from both documents and queries based on provided stopword list after converting all text into lowercase. Furthermore, tokenization, punctuation removing and stemming were sequentially implemented by using *nltk. RegexpTokenizer* and *nltk. PorterStemmer*.

4 Basic VSM Model

After preprocessing, the basic SVM was Implemented. Firstly, six different TFIDF schemas were used to represent 3,237 queries and 5,371 documents based on the preprocessed text. Then the distance between each query and each document was measured by cosine similarity, based on which, documents of each query were ranked from high relevance to low relevance. Finally, MAP and nDCG score of the whole queries and documents set were computed according to the actual ranking and the computed ranking. The TFIDF scheme with best MAP and NDCG score would be chosen as our final weighting method for speed-up model.

4.1 Weighting Schema–TFIDF

TFIDF	Document term weighting	query term weighting
Schema 0	$\frac{1+\log f_{t,d}}{1+\log(max_t f_{t,d})}.\log\left(\frac{N}{n_t}\right)$	$\frac{1+\log f_{t,d}}{1+\log(max_t f_{t,d})}.\log\left(\frac{N}{n_t}\right)$
Schema 1	$\frac{1+\log f_{t,d}}{1+\log(max_t f_{t,d})}.\log\left(\frac{N}{n_t}\right)$	$f_{t,d}$
Schema 2	$f_{t,d}.\log\left(\frac{N}{n_t}\right)$	$(0.5+0.5\frac{f_{t,d}}{max_t f_{t,d}}).\log\left(\frac{N}{n_t}\right)$
Schema 3	$1+\log f_{t,d}$	$\text{Log}(1+\frac{N}{n_t})$
Schema 4	$(1+\log f_{t,d})\log\left(\frac{N}{n_t}\right)$	$(1+\log f_{t,d})\log\left(\frac{N}{n_t}\right)$
Schema 5	$\frac{1+\log f_{t,d}}{1+\log(max_t f_{t,d})}.\log\left(\frac{N}{n_t}\right)$	0 or 1

Figure 1: TFIDF Schemas

The most important part in SVM is to determine the weight of terms in queries and documents.TDIDF, short for term frequency-inverse document frequency, is often used as a weighting factor in IR. There are three main factors to decide the term weight in TFIDF: frequency of a term in a document, the length of the document which contains the term and the document frequency of a term. Different considerations of these three factors produce various TFIDF schemes (Paik, 2013). In our project, six different TFIDF schemes were tested in basic model as above Figure 1.

4.2 Distance Measurement

Cosine similarity, as the most commonly used distance measurement in high-dimensional positive spaces, was selected in our project. Cosine similarity is a measure of distance between two non-zero vectors, which measures cosine of the angle between them. Due to it is particularly used in positive space, the outcome is between 0 and 1. The cosine of 0° is 1, and the bigger the angle is, the smaller the cosine value is. So it is a judgment of orientation and not magnitude: two vectors with the same orientation have a cosine similarity of 1, two vectors with different orientation have lower similarity.

5 Speed up models

Basic SVM has a good performance in document ranking but its time cost is too expensive. Therefore SVM speed up models are proposed to shorten the time while trying to not lose too much performance. Normally, it can be realized by reducing the total number of similarity comparison or reducing the the set of query and document terms. In our project, three common speed up models were used based on the basic model: tired index, random projection and pre-clustering.

5.1 Tiered Index

Tiered index is an approach which splits posting list into several tiers. Top K relevant documents can be obtained from tiers. Since the size of considered documents decreases, the responding retrieval time is minimized (Panigrahi and Gollapudi, 2013). Furthermore, a posting list is a list of documents that contains a term, and the number of posting lists is the number of terms in a query. Since IDF is the same for each term in different documents, the measurement of splitting posting list is according to TF score of terms in documents. Additionally, there are two methods to divide whole documents into several tiers, one is to divide it with the same length of tiers, and another is to divide it according to some thresholds.

There are four important parameters in this approach: the number of tiers, the documents number of each tire, Top K value, and the required percentage of query terms a document should contain.

Number of tiers. The number of tiers is flexible and decides the quantity of returned documents, which also influences the responding time. In this study, in order to figure out influence of number of tiers, value of tiers was set from two to nine

Documents number of each tire. Documents number of each tire can be flexible and decided by threshold or be fixed. In here, the first dividing method was chosen, because each query term has different frequency domain of its corresponding posting list, it is not scientific to use the same threshold value to split all posting lists. The first dividing method were implemented. The number of documents is fixed here.

Required percentage value . Since it is nearly impossible that a document contains all query terms, the required percentage of query terms a document contains needs to be set. The document could be returned if it contains at least required percentage terms. The domain of percentage value was from 0.01 to 0.09 in the research.

Top K value . Top K value is the lower bound value of number of return documents. Start from the first tier, if the total number of documents returns in this tier is not smaller than K value, these documents are the final return relevant documents, otherwise, this tier and the next tier would be merged, and checking number of relevant documents. Iteratively, the top relevant documents are returned. There were two kinds of set of top documents might be returned, one was the number of documents was satisfied the requirement, and another was that the number of documents was not enough, but all tiers were calculated. After number of testing with different value of K, the tendency of time consumption was increasing with K increasing. K value was set as 50 in this study.

5.2 Pre-Clustering

In procedure of basic VSM-based ranking model, it has a large amount of similarity computations by comparing each single query with every document linearly. In order to reduce the time-consumption, a new speed-up ranking approach was achieved by integrating VSM and document clustering model. Document clustering is a text mining technique which groups different document based on their similarity degree. (Nurminen et al., 2005). Apart from pre-clustering model, document clustering is also widely applied in text browsing and search results clustering fields (Yang, 2003). To test the influence of different clustering model on both time and precision of ranking system, following two cluster algorithms were used:

Single-path. After the TFIDF vector computation, single-path clustering firstly randomly selected k document as leaders by using *np.random*. Next estimated cosine similarity of each document with each leader. Afterwards every document was assigned to a specific closest leader cluster based on similarity results. k is a variable of leader size which depends on document size. Due to the inherent uncertain impact on ranking performance of changed leader size and changed random selection result, the project tested a bunch of leader size with average ten random selections for each size. The detailed size setting is showed in section 7.

K-means. Same as single-path approach, k-means also firstly randomly selected k documents as initial leaders based on the TFIDF vector representation then assigned each document to closest leader's cluster. Apart from the single-path, K-means continued to optimize clusters. Based on initial cluster results, K-means next recurrently reselected the centroid point of each cluster as new leaders and restructured clusters by recomputing previous document distances with new leaders. Finally, the clustering restructure process stopped until no changes in leaders as well as clusters. There are two parameters in this method: initial leader seed and the leader size. The initial leaders were random selected using *np.random*. But the more important uncertainty for this approach is leader size which directly impact the number of documents needed to be ranked. For both approaches, the leader size k was tested 47 different values from 50 to 5000.

5.3 Random Projection

Dimensionality reduction techniques. Beyond the basic model, dimensionality reduction of document and query vector is a common way to shorten the retrieval time. This technique reduces the number of components of a data set by representing the original data as accurately as possible with fewer features. The goal is to produce a more compact representation of the data with only limited loss of information so that the storage and speed up the model is reduced. The ultimate measure of success for any dimensionality reduction technique is the impact on the retrieval performance. There are several well known dimensionality reduction techniques: singular value decomposition (SVD), principal component analysis (PCA), independent component analysis (ICE), Non-negative matrix factorizations (NMF) and so forth (Berka and Vajteršic, 2014).

Random Projection. Locality-sensitive hashing(LSH) was used in our project to implement dimensionality reduction. The key idea is to hash the points using several hash functions so as to ensure that, for each function, the probability of collision is much higher for objects which are close to each other than for those which are far apart (Datar et al., 2004). Several methods could realize LSH, for example, bit sampling for hamming distance, min-wise independent permutations and random projection. And random projection was selected due to its simplicity.

The random projection steps for documents are:

i Choose a set of M random vectors $\{r_1, r_2, \ldots, r_M\}$ based on the normal(Gaussian) distribution of each term in every document (original vector length: $\mid V \mid$)

ii For each document TF-IDF vector d do:

Compute d and each random vector r:$\theta(r, d) = \sum_i^{|V|} r_i * d_i$

Hash each inner product: $h(d, r_k) = 1$ if $\theta(d, r_k) > t$ (threshold) otherwise 0

iii Compute a new vector: $d' = [h(d, r_1), h(d, r_2), \ldots, h(d, r_M)]$ and the number of selected random vectors, M, is the dimensionality of new vectors.

The random projection for the query is the same with that for documents.

Factors Consideration. There were three important factors in random projection: the number of random vectors M, threshold t and similarity measurement.

- For M, 8 values from 10 to 3000 were tested and they are 10, 100, 300, 500, 1000, 2000, 2500 and 3000.

- For threshold t, two methods were used. One assigned the median of the whole vector to t, respectively to the documents and queries. Another, for each document and query, assigned the median of its own vector to t. That, implies, different documents and queries have different t.

- For similarity measurement, having considered the vector structure after compared with threshold t which contains only 0 and 1, except cosine similarity, hamming similarity was selected. Hamming similarity measured the number of equals components between two vectors.

6 Evaluation Metrics

Two widespread measurements for information retrieval are used here, MAP and NDCG. Due to these two approaches discard similarity score and use rank, before evaluating the performance of ranking model, the positions of label of relevant level from dataset are rearranged according to the order of sorted cosine similarity score order to evaluate.

6.1 Mean Average Precision

First measurement is Mean Average Precision (MAP), MAP is the mean value of average precision of queries, and the result of MAP is only influenced by the rank of relevant level, not the relevant score. Average precision is sum of each precision on the rank position which represents the corresponding document is relevant to the query. Relevant documents in low rank position would get penalty. (R_{jk} is the rank position of the kth relevant document for the jth query. m_j is the number of relevant documents in the jth query. $\mid Q \mid$ is the quantity of all queries.

$$MAP = \frac{1}{\mid Q \mid} \sum_{j=1}^{|Q|} \frac{1}{m_j} \sum_{k=1}^{m_j} P(R_{jk}) \quad (1)$$

Since the relevant level of queries and documents in the dataset is multi-level, but MAP is used for binary-level. Thus, before applying MAP, turning multi-level into binary-level is necessary. Remaining relevant levels of irrelevance and relevance, and discards the relevant grade.

6.2 nDCG

Furthermore, another approach is using Normalized Discounted Cumulative Gain (NDCG) to study the model performance, which can be achieved with relevant multi-level. Since the number and relevant degree of documents are different with different query, it is hard to compare the DCG scores. The solution is a normalized function of DCG using IDCG (idea DCG).

$$nDCG = \frac{DCG(k)}{IDCG(k)} \quad (2)$$

DCG is sum of relevant grade of documents at its position, but the function to calculate it can be different, thus there are several DCG function can be chosen for different measurement purpose (Wang et al., 2013). For instance, two DCG functions are chosen in this study. First DCG function is:

$$nDCG = \sum_{i=1}^{k} \frac{rel_i}{\log_2 (i+1)} \quad (3)$$

This above function more emphasizes penalty, and high relevant degree of documents on high rank position can contribute more, and rel_i is relevant degree of the ith document. Another function cares about all relevant documents.

$$nDCG = \sum_{i=1}^{k} \frac{2^{rel_i} - 1}{\log_2 (i+1)} \quad (4)$$

7 Result Analysis

7.1 Basic VSM

Different weighting schema affects the ranking performance of space vector model. According the Table 1, by combined considering MAP, nDCG and time, schema 0 and 4 have similar results. To be consistent and comparable, schema 0 was adopted by following speed-up models. Meanwhile, due to the highly similarity of nDCG1 and nDCG2 results, only the nDCG1 was selected from two nDCG metrics to in latter speed up models.

Metrics	S0	S1	S2	S3	S4	S5
MAP	0.13	0.11	0.11	0.11	0.13	0.09
nDCG1	0.49	0.46	0.48	0.46	0.49	0.43
nDCG2	0.49	0.46	0.48	0.0.46	0.49	0.43
Time	518	1158	499	555	507	1092

Table 1: Comparison of different TFIDF

7.2 Tiered Index

There are two variables to be discussed in this section. In order to analysis the influence of each variable, single variable method was used.

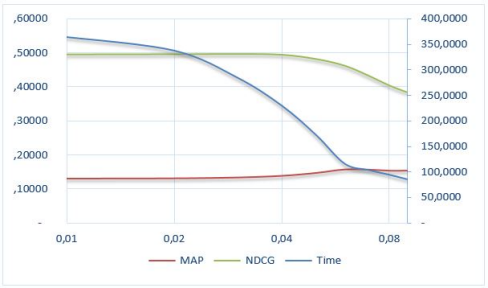

Figure 2: percentage value

i. *Required percentage value.* In Figure 2, values of tiers is set as three. This figure shows thats time is decreasing with the increasing of percentage percentage . Since required percentage of query terms a document contains might influence the quantity of return documents of each query, and the higher percentage value could decrease the number of return relevant document, which would increase the ranking speed.
For evaluation value, MAP shows a slight continuously increasing trend, while nDCG is decreasing. The reason of MAP increasing is that the relevant documents in low rank position are pruned. In contrary, the second nDCG function does not have penalty, thus the result of it decreases.

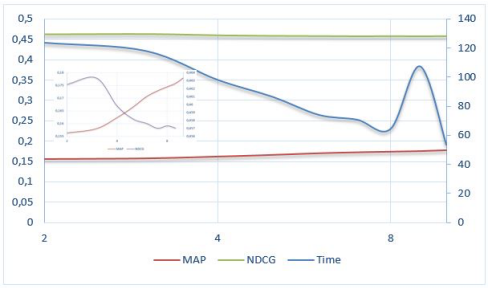

Figure 3: tiers number

ii. *Tiers number.* In Figure 3, percentage value is set as 0.06. The small chart in this figure shows details of evaluation results tendency. As blue line shown,the tendency of time consumption is decreasing, and there is a peak wave in the curve. Since the requirement of return document quantity is not smaller than K value, the fewer documents might be returned with the more tiers, thereby the time decreases. The next iteration is implemented and add more relevant documents, thus the return documents could be more than the quantity of document when tiers value equals to 8.

7.3 Pre-Clustering

During pre-clustering stage, two approaches of single-path and k-means were implemented to estimate the trade off liability of clustering-based ranking model on time consumption and ranking precision. After large number of experiments, the results are summarized as Figure 4 and Figure 5.The right y-axis presents time interval and the right one presents MAP and nDCG scope.

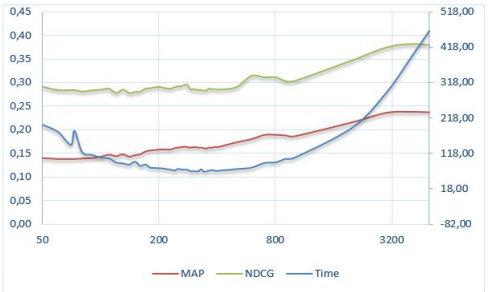

Figure 4: Performance of pre-clustering with single-path

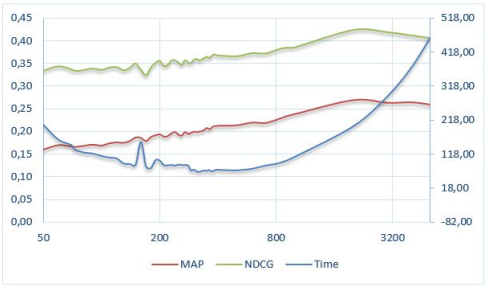

Figure 5: Performance of pre-clustering with k-means

i. By comparing two charts, it could be induced that K-means has a general better performance than basic single- path clustering even though it takes slightly more time. According numerical comparison with average of MAP, nDCG and time between two approaches, K-means shows around 20% improvement on both MAP and nDCG but 1% increasing in time consumption.
ii. From the results of both two approaches, it shows that with the increasing of cluster amounts, the MAP value is almost linearly increased as well. The average number of documents in each cluster is decreased which raise the AP of each query.
iii. As for NDCG, it presents a U-shape development tendency with increasing of number of cluster and finally peaks at more than 2000 clusters in both approaches. For smaller number of clusters, it losses less correct documents. Two cluster will loss smallest correct documents. This might caused the left high NDCG peak. For larger number of clusters, it contains fewer documents inside which have higher probability to be correct documents. This might explain the right peak of nDCG.
iv. Moreover, the time consumption of both two methods shows a bigger U-shape tendency. Firstly, the time continuously drops until the number of cluster goes up more than around 400, afterwards, it increases until the end. Due to the documents amounts decreased in each cluster with increasing number of clusters, the number of similarity comparison among each query and target documents reduces correspondingly. This reduced the time consumption. However, the excessive clusters will increase the comparison between queries and leaders. Until the number of cluster equals to number of documents, the

pre-clustering losses its speed-up competition. The small time peak during decreasing trend might result from the possible unbalanced document clustering results

7.4 Random Projection

According to the three main factors mentioned before, the results are compared in following three aspects.

i. *Cosine and hamming similarity comparison.* As is shown in Figure 6, compared with cosine similarity, the retrieval time of hamming similarity is three times faster on average. And the performances (MAP and nDCG) are almost the same according to figure 7. For example, when the final vector length is 10, the retrieval time of cosine similarity is 302.7s, with 83.5s of hamming similarity, and both MAP and NDCG score are the same. Based on this finding, the hamming similarity was finally selected as the distance measurement in random projection.

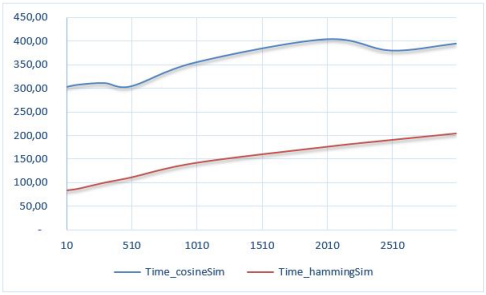

Figure 6: Time comparison of cosine and hamming similarity

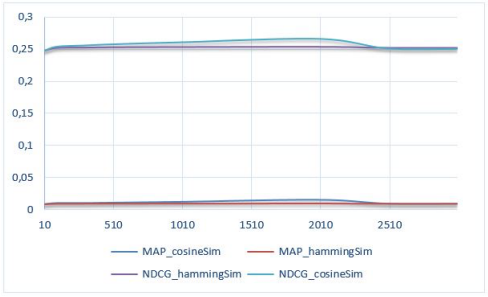

Figure 7: Performance comparison of cosine and hamming similarity

ii. *Thresholds comparison* . The result in first finding is based on one global threshold. As shown in Figure 8 and 9, the MAP and nDCG of choosing different thresholds is much higher than that of one global threshold. For example, when the vector length is 2000, the MAP and NDCG with one global t are 0.0096 and 0.2541 respectively and they are lower than 0.046 and 0.32 of different thresholds. That, implies, the performance of different thresholds is better than one global threshold overall.

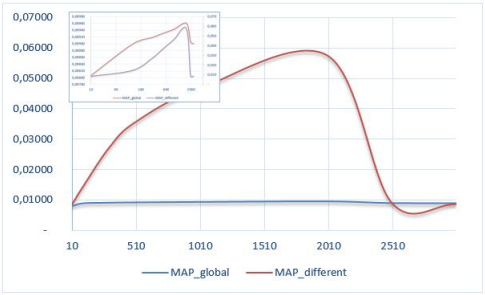

Figure 8: MAP comparison of one threshold and different thresholds

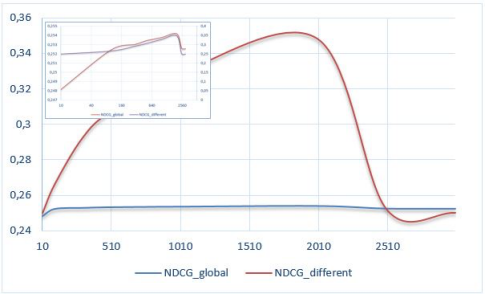

Figure 9: NDCG comparison of one threshold and different thresholds

iii. *Vector length comparison.* Additionally, from Figure 6, it is also obvious that, as the vector length grows from 10 to 3000 , the retrieval time grows from 83s to 204s using hamming similarity and the same trend with cosine similarity. From Figure 8 and 9, MAP and nDCG scores also increase overall. But as is shown in the top left corner of the two charts, there exits a same peak when the vector length is about 2000. After 2000, the performance starts to decrease as the vector length increases. The MAP and nDCG values in vector length of 2500 and 300 are both lower than that in 2000.

Overall, considered the retrieval time, MAP and nDCG, the best result in random projection so far is 184.89s, 0.057 of MAP and 0.348 of nDCG, with hamming similarity computation, different thresholds and vector length 2000.

Metrics	Basic	TI	PC	RP
MAP	0.13	0.16	0.20	0.06
MAP %	-	0.26	0.60	-0.54
nDCG	0.49	0.46	0.36	0.35
nDCG %	-	-0.06	-0.27	-0.29
Time/q	0.26	0.03	0.03	0.09
Time/q %	-	-0.88	-0.88	-0.63

Table 2: Comparison of Speed-up Models

7.5 Comparison

After the testing of three speed-up VSM models, a comprehensive performance comparison overview is summarized as Table 2. Where *Basic* is basic VSM model, *TI* is Tiered Index, *PC* is pre-clustering and *RP* is random projection. Regarding basic VSM model as the baseline, both precision and time improvement were calculated in percentage. The time per query was estimated by take the average running time of 200 random queries. In terms of Table 2, we have following findings:

i. All three speed-up models successfully reduced the time of ranking system. Tiered index and pre-clustering seeped up most portion of time consumption by 88%. Random project also achieved a good time reduction with 63%.

ii. From MAP perspective, pre-clustering approach got the highest score which improves basic SVM by 60%. However it is benefit from the dramatic cutting of number of documents. Random projection loss a lot MAP as expected because it shortens the vector length which will negatively impact on the similarity estimation meanwhile kept full documents collection.

iii. As for nDCG, Tiered index performed best. Pre-clustering has similar performance with random projection.

iv. In general, by mainly considering metrics of nDCG and time speed-up, tiered index has a best performance among three speed-up models.

8 Conclusion

In this research, tiered index, pre-clustering and random projection have successfully been implemented to decrease time consumption of ranking system. In terms of comparison result, tiered index has the outstanding performance regarding to time consumption speed up and nDCG score. There is a hardware limitation should be considered, which is the time consumption might be impacted by facilities condition. In the future study, the hybrid model of combination of individual approach could be researched to decrease ranking time and increase evaluation performance.

References

Tobias Berka and Marian Vajteršic. 2014. Dimensionality reduction for information retrieval using vector replacement of rare terms. In *Data Mining for Service*, pages 41–60. Springer.

Vera Boteva, Demian Gholipour, Artem Sokolov, and Stefan Riezler. 2016. A full-text learning to rank dataset for medical information retrieval. In *Proceedings of the European Conference on Information Retrieval (ECIR)*. Springer.

Mayur Datar, Nicole Immorlica, Piotr Indyk, and Vahab S Mirrokni. 2004. Locality-sensitive hashing scheme based on p-stable distributions. In *Proceedings of the twentieth annual symposium on Computational geometry*, pages 253–262. ACM.

Miika Nurminen, Anne Honkaranta, and T Karkkainen. 2005. Extminer: Combining multiple ranking and clustering algorithms for structured document retrieval. In *Database and Expert Systems Applications, 2005. Proceedings. Sixteenth International Workshop on*, pages 1036–1040. IEEE.

Jiaul H Paik. 2013. A novel tf-idf weighting scheme for effective ranking. In *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval*, pages 343–352. ACM.

Debmalya Panigrahi and Sreenivas Gollapudi. 2013. Document selection for tiered indexing in commerce search. In *Proceedings of the sixth ACM international conference on Web search and data mining*, pages 73–82. ACM.

Yining Wang, Liwei Wang, Yuanzhi Li, Di He, and Tie-Yan Liu. 2013. A theoretical analysis of ndcg type ranking measures. In *Conference on Learning Theory*, pages 25–54.

Kiduk Yang. 2003. Combining text-, link-, and classification-based retrieval methods to enhance information discovery on the web.