Efficient Semantic Ranking for Top-K Document Retrieval Based on SG-Reversed Index

M.Uma Maheswari[1]
[1]Research Scholar
Department of Computer Science
Bishop Heber College, Tiruchirappalli, TN
India 620017
umamanohar89@gmail.com

Dr.J.G.R.Sathiaseelan[2]
[2]Associate Professor & Head.
Department of Computer Science
Bishop Heber College, Tiruchirappalli, TN
India 620017
jgrsathiaselan@gmail.com

Abstract- In Information retrieval, the main issue is discovering the top-k documents from huge repositories. An efficient top-k retrieval algorithm is needed for the real time applications such as document search, online advertising and DBMS. Efficient ranking answers the user queries in the repository that continues to challenge the researchers as the size grows proportional to time, and the ranking metrics become more complicated. Ranked document retrieval is usually solved with some variant of a simple structure called an inverted index. This paper proposes an efficient semantic ranking for top-k document retrieval based on SG-Reversed Index. Given a keyword query, the main goal is to find the top-k most relevant documents. This paper proposes two approaches baseline search and semantic search for retrieving and ranking the top-k documents using single or multiple term queries with Boolean keywords. In baseline search, the simple reversed index was scanned each keyword in the query to compute the ranking of all documents in the document collection. The semantic search method computes the semantic relatedness between query keyword and document collection to retrieve top-k high ranked documents. In addition to this Semantic Graph based reversed index is formed for quick retrieval of documents. Experiments are conducted on real time text datasets to verify the effectiveness and efficiency of the proposed approaches in terms of precision, recall and f-measure.

Keywords: Information retrieval, Top-k Retrieval, Inverted Index, Document Search, Semantic Ranking

1 Introduction

The rapid growth of documents, web pages and other types of textual content pose a great challenge to modern content management systems. In an information retrieval, a document search involves determining which documents are relevant to a query. Traditionally, the search process starts with a query and a corpus of documents, and compares the words in the query with the words in a document. A scoring algorithm assigns scores to the documents in the corpus based on which words in the query appear in the various documents, and with what frequency. The documents are then ranked based on their scores, and the results of the search are presented. Traditional scoring technique includes term-based scoring i.e., term vector space modeling. It can be effective, but they have their limitations. In particular, there may be some information that is not part of the query itself, but nonetheless suggests what a user is looking for.

Most of the existing document retrieval system still relay on standard retrieval models that treat both document and query as a set of unrelated terms. These statistical models have the advantages of being simple, scalable and computationally feasible but they do not offer accurate and complete representation. These models ignore semantic and contextual information in the retrieval process.

To improve the relevance in document retrieval, a semantic based conceptual graph was proposed in [1]. An ontology based retrieval model was used in [2] for the exploitation of environmental sciences domain ontology's and knowledge bases, to support semantic search in document repositories.

This paper proposes semantic based document retrieval based on SG-Reversed Index. A traditional data structure called inverted index which given a term provides access to the list of documents that contain the term. The inverted index is the list of words and the documents in which they appear [3]. The main motive of an inverted index structure is to allow fast searching of text with increased processing speed when a document is added to the original database [4] but ignored a lot of words associated with the semantic, and limited the ability to provide the effective retrieval [5].

To overcome this problem, this paper generate Semantic Graph (SG) based revered index for fast document search and effective semantic document retrieval. The inverted index is constructed from extracted features. Based on this SG-Reversed Index is constructed. This paper proposes an efficient document search approach that retrieves top-k related documents from the document collection based on baseline and semantic search method. In baseline search, the simple reversed index was scanned each keyword in the query to compute the ranking of all documents in the document

collection. The semantic search method use SG-Reversed Index and computes the semantic relatedness between query keyword and document collection to retrieve top-k high ranked documents.

The division of the paper is described as follows. Section 2 provides the related work of different document retrieval techniques. Section 3 explains proposed semantic top-k document retrieval and Section 4 discusses the performance analysis of proposed results. Finally Section 5 summarizes the work which has been done.

2. Related Work

Many document retrieval algorithms have been proposed so far in Information Retrieval. This section gives an overview of related work on document search and rank techniques.

There are two main approaches for solving the top-k retrieval problem: Inverted Index and Posting List. Inverted indexes are used almost exclusively in practice by real-world search engines such as Apache Lucene [18]. Inverted indexes store all occurrences of a term in a document in posting lists. Decades of research have yielded very compact and fast indexes using this technique. Nevertheless, there are some inherent limitations of the inverted index approach: In order to support phrase queries efficiently, additional information needs to be stored alongside the postings, and heuristics are used to ensure that tokens appear together and in the correct order in the resulting documents.

The top-k document retrieval is the one of the most important problem in information retrieval. Muthukrishnan [19] introduced an optimal time index for document listing in 2002. The solution uses a range minimum query data structure to enumerate all distinct document IDs in a lexicographic range. Hon et al. [6] presented a solution for the top-k document retrieval problem for the case when the relevance measure is tf(P,d) (the number of times P occurs in d). Culpepper et al. [7] built on an improved document listing algorithm on wavelet trees [8] to achieve two top-k algorithms, called Quantile and Greedy, which use the wavelet tree alone. Navarro [9] use [10] succinct structure on top of a wavelet tree, but instead of brute force [9] use a variant of Culpepper et al.'s [7] method to find the extra candidate documents.

Konow et al., [11] proposed Faster Compact Top-k Document Retrieval, the main idea of this work is to replace suffix tree sampling by frequency thresholding to achieve compression. Brisaboa et al., [12] introduced the K^2–treap to solve weighted top-k range queries faster and more space-efficiently in practice. Gog & Navarro [13] simplified the implementation of [11] by introducing a new mapping of suffix array ranges to coordinate ranges of the grid. Gagie et al., [14] show how document listing, top-k retrieval and document counting can be solved while exploiting the properties of repetitive collections. They introduce the interleaved LCP array and pre-computed document lists. Both concepts might also be applicable to top-k retrieval on non-repetitive collections.

Dimopoulos et al., [15] proposed Optimizing top-k document retrieval strategies for block-max indexes. The *Topic Enhanced Inverted Index* (TEII) was proposed in [16] to incorporate the topic information into the inverted index for efficient top-*k* document retrieval. It explores two different types of TEIIs: Incremental and hybrid. In [17], a new time/space trade-off for different top-k indexes was proposed. This is achieved by sampling only a quantile of the postings in the original inverted file or suffix array- based index.

A standard approach to Information Retrieval (IR) is to model text as a bag of words. Alternatively, text can be modeled as a graph, whose vertices represent words, and whose edges represent relations between the words [20]. Graph models have the capability of capturing structural information in texts but they do not take into account the semantic relations between words. Semantic relation is specified using Thesaurus Graph and concept Graph. In treasure Graph vertex denotes terms and edge denotes sense relations for example synonymy and antonym. Conceptual Graph is constructed from text document. Word net and Verb net is used to find the semantic roles in a sentence and using these roles conceptual Graph is constructed raw text are pre-processed and disambiguated nouns are mapped to WordNet concepts [21]. Paul et al., [22] present a scalable approach for related-document search using entity-based document similarity. The semantic similarity approach exploits explicit hierarchical and transversal relations.

In this paper, a semantic ranking model was proposed to retrieve top-k documents based on SG-Reversed Index. It uses baseline and semantic search method for top-k document retrieval.

3. Proposed Methodology

In this section the proposed top-k document retrieval based SG-Reversed Index model was explained. Figure 1 shows the architecture of proposed semantic top-k document retrieval.

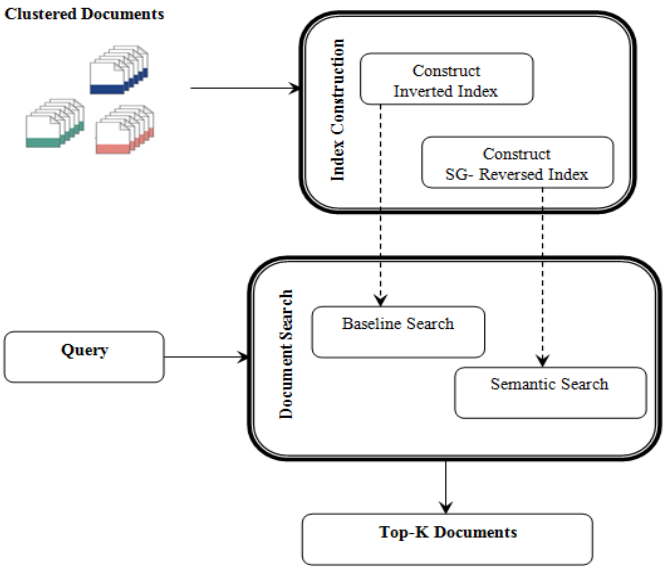


Figure 1 System Architecture

The main components of the system model are as in the following:

Index Construction

There are two types of indexes are constructed, one is traditional inverted index and another one is SG-Reversed Index i.e. Semantic Graph based Reversed Index.

Inverted Index

An inverted index of document is an index data structure used for storing content of original document in compressed form. The main motive of an inverted index structure is to allow fast searching of text with increased processing speed when a document is added to the original database. It is possible that the inverted data may be the database file itself, rather than index of that file. It is the most popular data structure used for document storing and accessing the systems, used on a big scale for example in Google [4]. In this paper the words inverted index and reversed index are interchangeable.

Figure 2 shows the sample inverted index.

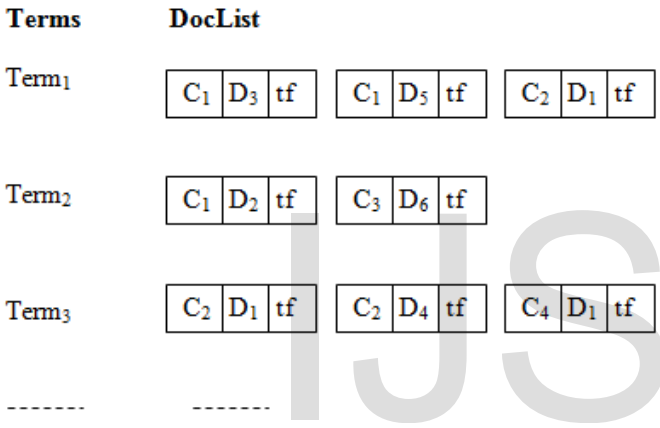


Figure 2 Sample Inverted Index

In this figure 2, the $Term_1$ occurs in D_3, D_5, D_1 within the cluster C_1, C_1, C_2

$Term_2$ occurs in D_2, D_6 within the cluster C_1, C_3 and so on.

Algorithm 1: Reverse Index Construction
Input: Clustered Documents CD, Terms T
Output: Reversed Index RI

1. For each term $t_i \in T$
2. For each cluster $c_j \in CD$
3. For each document $d_k \in$ cluster c_j
4. Map term t_i to document d_k
5. If $t_i \in d_k$ then
6. Compute term frequency tf
7. Append $\left(t, cls_{Id}, \langle doc_{Id}, tf \rangle\right)$ to RI
8. EndIf
9. EndFor
10. EndFor
11. EndFor

Algorithm 1 explains the simple reversed index. The input to the algorithm is clustered documents and terms extracted from each documents. For each term in a cluster, a pair consisting of the document id and the term frequency is created. Each pair denoted $\left(\langle doc_{Id}, tf \rangle\right)$ in the algorithm represents an individual term in cluster. The reversed index structure includes $\left(t, cls_{Id}, \langle doc_{Id}, tf \rangle\right)$ where t = term, cls_{Id} = Cluster Id, doc_{Id} = Document Id and tf = frequency of term t.

SG-Reversed Index

In general, a semantic graph is "a network that represents semantic relationships between concepts. It is a directed or undirected graph consisting of vertices, which represent concepts, and edges, which represent semantic relations between concepts." In this paper, the reversed index is constructed based on the semantic graph. Algorithm 2 explains the SG-Reversed Index construction.

Algorithm 2: SG- Reverse Index Construction
Input: Clustered Documents CD, Terms T
Output: Reversed Index SG-RI

1. For each term $t_i \in T$
2. Create Node n_i with node label term t_i
3. Extract all semantic concepts SC_i of term t_i
4. For each semantic concept $sc_j \in SC_i$
5. For each cluster $c_j \in CD$
6. For each document $d_k \in$ cluster c_j
7. Map term t_i and sc_j to document d_k
8. If $t_i \in d_k$ || sc_j is semantically related to d_k then
9. Create Node n_j with node label as sc_j
10. Compute semantic relatedness SR between sc_j and d_k
11. Create a Edge between n_i and n_j;
12. Assign SR as the edge value
13. Append $\left(t, SC, cls_{Id}, \langle doc_{Id}, SR \rangle\right)$ to SG-RI
14. EndIf
15. EndFor
16. EndFor
17. EndFor
18. EndFor

In this algorithm, each term extracts all semantic concepts. Each concepts map with document collection. A semantic relatedness is computed between concept and document. The terms and semantic concepts are considered as node, and an edge will be a document id with semantic relatedness of documents. The SG-Reversed index is used for semantic based document search.

Document Search

In this paper, the documents are searched based on base line and semantic search method. Algorithm 3 explains the top-k document retrieval.

Algorithm 3: Top-k Document Retrieval
Input: Reversed Index RI, SG-RI, Query Q, k
Output: Top –k Retrieved Documents

1. Initialize docList and scrList
1. For each $q \in Q$
2. If $q \in$ RI then // *Base Line Search*
3. Extract $\left(cls_{Id}, \langle doc_{Id}, tf \rangle\right)$ from RI for the term q

4. If ($doc_{Id} \notin$ docList)
5. Add doc_{Id} to docList and tf to scrList
6. Else If ($doc_{Id} \in$ docList)
7. Update scrList
8. EndIf
9. Else If q $\in$ SG-RI then // *Semantic Search*
10. .Extract $\left(SC, cls_{Id}, \langle doc_{Id}, SR \rangle\right)$ from SG-RI
11. Extract all the neighboring nodes of q
12. Find the doc_{Id} of neighboring nodes
13. If ($doc_{Id} \notin$ docList)
14. Add doc_{Id} to docList and SR to scrList
15. Else If ($doc_{Id} \in$ docList)
16. Update scrList
17. EndIf
18. EndIf
19. EndFor
20. Sort docList according to high scrList
21. Return Top-K Documents from docList

An efficient document search algorithm was proposed to retrieves the top-k related documents from the document collection based on baseline and semantic search method. In baseline search, the simple reversed index was scanned each keyword in the query to compute the ranking of all documents in the document collection. The semantic search method use SG-Reversed Index and computes the semantic relatedness between query keyword and document collection to retrieve top-k high ranked documents.

For each term q in a Query Q this searches the related documents. If the query term q contains in Reversed Index RI, the baseline search will be started to find the documents. It extracts all the documents i.e., doc_{Id} and corresponding term frequency of doc_{Id} from RI and stores into docList (contains list of doc_{Id}) and scrList (contains list of score value for corresponding doc_{Id}).

If the query term q contains in SG-RI, the semantic search will be started to find the documents. It extracts all the neighboring nodes, which are semantically related to query term q. It finds the doc_{Id} and corresponding semantic relatedness of doc_{Id} from SG-RI and stores into docList and scrList. The scrList was sorted into descending order and high score i.e., top-k documents are returned to user.

4. Experimental Result

This paper uses four evaluations metrics in the field of information retrieval: Precision, Recall, F-measure and MAP (Mean Average Precision) to test the efficiency of document retrieval.

Precision (P): The fraction of retrieved documents that are relevant.

$$P = \frac{|RET \cap REL|}{|RET|}$$

Where RET = Retrieved Documents and REL = Relevant Documents

Precision measures the system's ability to reject any non-relevant documents in the retrieved set.

Recall (R): The fraction of relevant documents that are retrieved

$$R = \frac{|RET \cap REL|}{|REL|}$$

Recall measures the system's ability to find all the relevant documents

The f-measure is calculated as,

$$F = 2 \times \frac{P.R}{P+R}$$

MAP (Mean Average Precision) is mean of Average Precision across multiple queries/rankings.

This paper uses the real time dataset BBC and 20 NewsGroups for experiments to analyze the efficiency of the proposed document retrieval system.

Table 1 shows the two text datasets which are used to investigate the performance of the proposed algorithm. The first data set is BBC, that contains 1000 random documents belongs to five groups i.e., five clusters (business, entertainment, politics, sport and tech). The second data set is 20News group, which contains 1000 random documents belongs to 10 groups.

Table 1 Dataset characteristic

Dataset	# of Docs	# of Clusters	# of Terms Extracted for Index Construction
BBC	1000	5	8862
20News Groups	1000	10	10243

Table 2 shows the index construction time for BBC and 20 News groups data set.

Table 2 RI and SGRI Construction Time

Dataset	SGRI	RI
BBC	26.396	221.536
20News Group	37.752	343.810

Figure 3 shows the reversed index and semantic graph based reversed index construction time for two dataset.

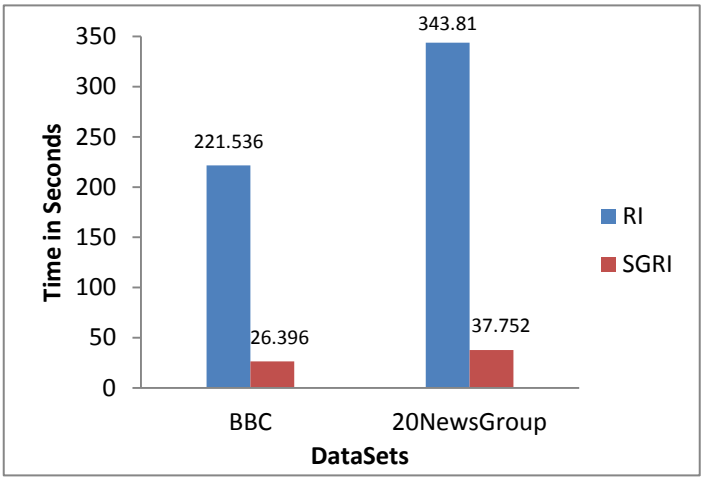


Figure 3 Index Construction Time for two Dataset

Table 3 Query Execution Time (Top-K=10 Results)

Data Set	Baseline	Semantic
BBC	31	35
20News Group	16	23

The query was executed with different number of keywords and different number of k values (i.e Top-k). Figure 4 shows the query execution time for query key word length = 2 and k=10.

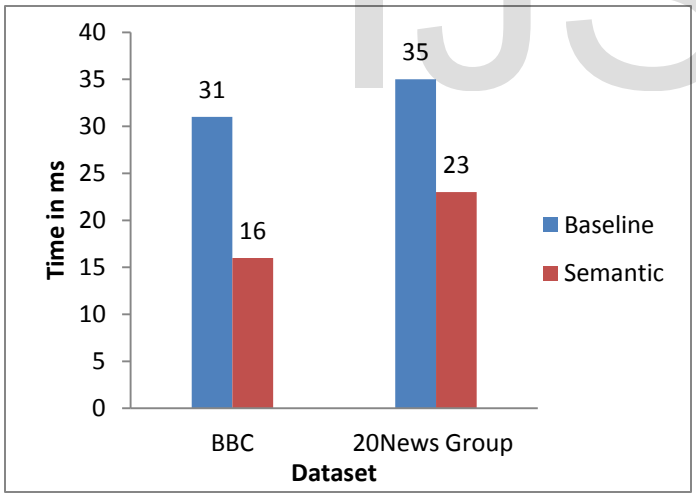


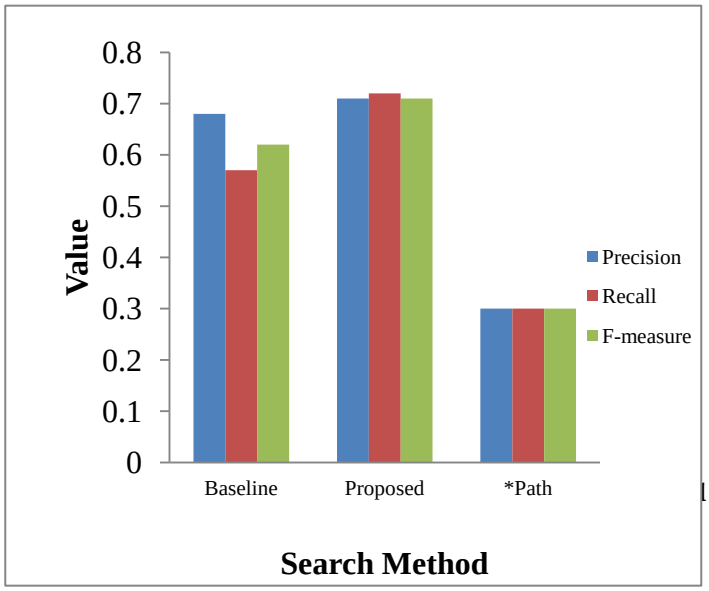
Figure 4 Query Execution Time for Top-k =10 results

Table 4 shows the precision, recall and f-measure comparison.

Table 4 Precision, Recall and F-Measure Comparison

Method	Precision	Recall	F-measure
Baseline	0.68	0.57	0.62
Proposed	**0.71**	**0.72**	**0.71**
*Path[23]	0.3	0.3	0.3

Figure 5 shows the comparison of precision, recall and f-measure for different search methods.



5. Conclusions

This paper presents a semantic top-k document retrieval model. This model first construct reversed index of preprocessed and extracted terms in a document collection. Based on this reversed index, an semantic graph based reversed index i.e., SG-RI was constructed. Next it proposes two kinds of search algorithms baseline and semantic search for retrieving top-k documents from document collection.

The baseline search uses RI and the semantic search uses SG-RI for document retrieval. The semantic retrieval method has exploited the advantages of the semantic concept to retrieve the relevant data. The experiment use real time data for the performance and its shown significant improvement in the ranking of retrieved documents.

REFERENCES

[1] Tanveer J. Siddiqui., Umashanker, Tiwary, (2006) "A Hybrid Model to Improve Relevance in Document Retrieval".Journal of Digital Information Management Vol. 4(1):73-81

[2] Sridevi, U. K., & Nagaveni, N. (2010). "Ontology based semantic measures in document similarity ranking" In Proceedings of the International Conference on Advances in Recent Technologies in Communication and Computing (pp. 482-486).

[3] Sodel Vázquez-Reyes et al., (2017) "The Use of Inverted Index to Information Retrieval: ADD Intelligent in Aviation Case Study", Trends and Applications in Software Engineering, Springer International Publishing, Pp. 211-220

[4] Sayali Borse, P. M. Chawan, (2016) "Inverted Index for Fast Nearest Neighbour", International Journal of Computer Science and Mobile Computing, Vol.5 Issue.6, June-2016, pg. 331-336

[5] S. Zhong and et al.,(2011) "A design of the inverted index based on web document comprehending", JCP 6(2011), no. 4, 664–670.

[6] W.-K. Hon, M. Patil, R. Shah, and S.-B.Wu (2010), "Efficient Index for Retrieving Top-k most Frequent Documents", Journal of Discrete Algorithms 8(4), 402-417 (2010)

[7] J. S. Culpepper, G. Navarro, S. J. Puglisi, and A. Turpin (2010) "Top-k ranked document search in general text databases", In Proc. 18th ESA , pages 194–205 (part II), 2010.

[8] T. Gagie, S. J. Puglisi, and A. Turpin. (2009) "Range quantile queries: Another virtue of wavelet trees". In Proc. 16th SPIRE, pages 1–6, 2009

[9] Navarro, G., Valenzuela, D.: (2012) "Space-efficient top-k document retrieval". In: Klasing, R. (ed.) SEA 2012. LNCS, vol. 7276, pp. 307–319. Springer, Heidelberg (2012)

[10] W.-K. Hon, R. Shah, and J. Vitter.(2009) "Space-efficient framework for top-k string retrieval problems" In Proc. 50th FOCS, pages 713–722, 2009

[11] R. Konow and G. Navarro. (2013) "Faster compact top-k document retrieval". In Proc. DCC, pages 5–17, 2013.

[12] N. Brisaboa, G. de Bernardo, R. Konow, and G. Navarro. K2-Treaps (2014) "Range Top-k Queries in Compact Space". In Proc. SPIRE, pages 215–226, 2014

[13] S. Gog and G. Navarro. (2015) "Improved single-term top-k document retrieval". In Proc. ALENEX, pages 24–32, 2015.

[14] Gagie, T., Hartikainen, A., Karhu, K., Kärkkäinen, J., Navarro, G., Puglisi, S. J., & Sirén, J. (2017). "Document retrieval on repetitive string collections". Information Retrieval Journal, June 2017, Volume 20, Issue 3, pp 253–291

[15] C. Dimopoulos, S. Nepomnyachiy, and T. Suel (2013), "Optimizing top-k document retrieval strategies for block-max indexes" In Proc. of the Sixth ACM International Conference on Web Search and Data Mining , 2013.

[16] Di Jiang, Kenneth Wai-Ting Leung, Lingxiao Yang, and Wilfred Ng. (2015). "TEII: Topic enhanced inverted index for top-k document retrieval", Knowledge-Based Systems 89 (2015), 346–358.

[17] Niklas Baumstark , Simon Gog , Tobias Heuer and Julian Labeit (2017),"The Quantile Index - Succinct Self-Index for Top-k Document Retrieval", 16th International Symposium on Experimental Algorithms, 2017

[18] D. Tsur.(2013) "Top-k document retrieval in optimal space". Information Processing Letters , 113(12):440–443, 2013.

[19] S. Muthukrishnan. (2002), "Efficient algorithms for document retrieval problems". In Proc. SODA, pages 657–666, 2002.

[20] Blanco, R., and Lioma, C. (2012), "Graph-based term weighting for information retrieval". Information retrieval 15(1):54–92

[21] Sonawane S and Kulkarni P. (2014), "Graph based Representation and Analysis of Text Document: A Survey of Techniques", Journal of Computer Applications, 96(19):1-8.

[22] Paul, C., Rettinger, A., Mogadala, A., Knoblock, C. A., and Szekely, P. (2016). "Efficient Graph-based Document Similarity". In Proc. of ESWC'16 . Springer.

[23]. E. Marx, K. Höffner, S. Shekarpour, A.-C. N. Ngomo, J. Lehmann, and S. Auer. (2016), "Exploring Term Networks for Semantic Search over RDF Knowledge Graphs", pages 249–261. Springer International Publishing, Cham, 2016