Layered Approach to Improve Web Information Retrieval

Jayant Gadge
Research scholar
Computer Tech. dept
VJTI, Mumbai

S.S. Sane
Professor
Computer Tech. Dept
VJTI, Mumbai

H.B. Kekre
Sr. Professor
MPSTME,
SKVM's MNIMS

ABSTRACT

The Web has become the largest available repository of data. The exponential growth and the fast pace of change of the web makes really hard to retrieve all relevant information. The crawling of web pages with speed for finding relevant set of document is perhaps the main bottleneck for Web search engines. There are many factors that affect web search such criteria are web data, user behavior and spam etc. For retrieval of information, many web information retrieval models have been proposed, studied and empirically validated. In these information retrieval models, the documents are typically transformed into a suitable representation to make the retrieval efficient.

In this paper, Information retrieval method is proposed based on the vector space model. The N-level vector space model is applied for information retrieval which is better suited for dynamic expansion of the web document set. Web document has semi-structured characteristics. The terms or keywords that are used for indexing purpose of document appear in special location such as title, subtitle, header, hyperlinks and so on. The content of these special locations represents important information in the web documents. Vector space model ignores the importance of these terms with respect to their position while calculating the weight of the indexing terms. In N-level Vector space approach, the importance of these terms with respect to their position is considered. The web document is logical divided in N-level considering the structure of web document and weights are assigned to terms based on their presence in different layer within the document.

General Terms

Web Information Retrieval

Keywords

Web Information Retrieval, Page Ranking, Vector space model, Layered Vector space approach.

1. INTRODUCTION

The Web has become the largest available repository of data. It is natural to extract information from it and web search engines have become one of the most widely used tools to get information access. The exponential growth and the fast pace of change of the web makes really hard to retrieve all relevant information. The crawling of web pages with speed for finding relevant set of document is perhaps the main bottleneck for Web search engines. Besides this, the characteristics of web search make it different from other types of search. The characteristics of the data and web user behavior affect the web search. The other characteristics that affect the web search are web data, web structure, user behavior, spam.[1][3].

Many different retrieval models have been proposed, studied and empirically validated. Web information retrieval models are mainly two categorized in two type, Traditional or classic information model and modern web information retrieval model.

In classic information retrieval model, the documents are typically transformed into a suitable representation to make the retrieval efficient. Classic information model aim to rank the documents based on the content of the collection.[7]

Modern web information retrieval exploits the link structure of the Web and log information of web. The links provide a positive critical assessment of a Web page's content which originates from outside of the control of the web page's author. The hyperlink structure is exploited by two of the most frequently used Web information retrieval methods HITS (Hypertext Induced Topic Search) and google PageRank algorithm [2][7].

In this paper, Information retrieval method is put forward based on the vector space model [9]. This model is referred as N-level vector space model. The N-level vector space model applied for information retrieval which is better suited for dynamic environment. The theoretical analysis and experimental result shows that suggested method improves the performance.

Web document has semi-structured for characteristics. The terms or keywords that are used indexing purpose of document appear in special location such as title, subtitle, header, hyperlinks and so on. The content of these special locations represents important information in the web documents. N-level Vector space approach, the web document is logically divided in N-level considering the structure of web document and weights are assigned to terms based on their presence in different layer within the document

In this paper, Literature review and all related work in web information retrieval is discussed in section 2. In Section 3, suggested N-level vector space model and algorithm is described. In Section 4, data structure required for implementation of N-level vector space model is presented. In Section 5, experimental results and graphs are discussed and in Section 6, the conclusion is presented.

2. LITERATURE REVIEW

The task of an information retrieval system is to identify relevant documents based on a user's information need. Over the past decades, many different retrieval models have been proposed, studied and empirically validated. Recent study shows different challenges in web search and information retrieval. These challenges are listed below.

2.1 Challenges in Web Information Retrieval

Recent work on searching the Web includes the methods that can be categorized to address following challenges listed below.

- Keeping the index fresh and complete, including hidden content.[6]
- Identifying and removing malicious content called spam. Common method of spamming involves placing additional keywords in invisible text in the web page so that pages potentially match many more user queries, even if the page is really irrelevant to the queries.
- Identifying content of good quality. The Web is full of low quality content i.e. syntactic and semantically, including noisy, unreliable and contradictory data.[6]
- Exploiting user feedback, either from explicit user evaluation or implicitly from Web logs[7]. The implicit information given be the authors of web pages in the form of several conventions used in HTML design.
- Detecting duplicate hosts and content, to avoid unnecessary crawling.
- Distinguishing the information need: informational, navigational, or transactional. It is estimated that less than 50% of the queries are of informational[1]
- Improving the query language, adding the context of the information needed.[5]
- Improving ranking, in particular to make it dependent on the person posing the query [5]. Relevance is based on personal judgments, so ranking is based on user profile or other user based context information.

In order to make information retrieval to be efficient, the documents are typically transformed into a suitable representation. There are several representations such as Boolean Retrieval model, Fuzzy Set model, Extended Boolean model, Vector Space model, Latent Semantic indexing model.

Boolean model [8] is based on set theory and Boolean algebra. The document is represented as set of terms and queries are as Boolean expressions formed using terms. Boolean model retrieves the document if there is exact match between query and set of document. Sometimes it leads to either too many or too few retrieved documents. This model is easy implement. This model has no provision for ranking documents and assigning importance factors or weights to query terms.

Fuzzy set model [3], documents and queries are presented by sets of index terms. The matching between document and query is approximate from start. This vagueness is modelled using fuzzy framework. In Fuzzy set model, the procedure to compute the document relevance to a query is analogous to the procedure adopted by the classic Boolean model. The only difference is that instead of using the Boolean sets, the Fuzzy set is used.

The Latent Semantic Indexing [10] information retrieval model builds upon the prior research in information retrieval and using the singular value decomposition [10]. It reduces the dimensions of the term-document space and attempts to solve the synonymy and polysemy problems that affect automatic information retrieval systems. LSI explicitly represents terms and documents in a rich, high-dimensional space, allowing the underlying semantic relationships between terms and documents to be exploited during searching. The process of matching documents could be based on concept matching instead of index term matching. The main idea in this model is to map each document and query vector into a lower dimensional space which is associated with concepts.

3. N-LEVEL VECTOR SPACE MODEL

In vector space model, use tf-idf weight as a statistical measure to evaluate how important a term is to a document in a collection. The importance increases proportionally to the number of times a term appears in the document but is offset by the frequency of the word in the collection. The baseline method for computing the weight of a term in a document is to count the number of times the term occurs in the document. This is referred as term frequency (tf). Term frequency does not exploit structural information present in the web page. For exploiting web page structure, terms frequency as well as position of term is also considered. This is referred as feature frequency.

The term, that appearing in the special locations such as title, hyperlinks, body and paragraph represents more important information in the web document. In N-level vector space model, the document is logically divided in three layers namely Title region, hyperlink region and body region and weights are assigned to terms based on their presence in different layer within the document.

Let $D = \{ D_1, D_2, D_3, \dots\dots\dots\dots D_n \}$ be the Document set

Let tf_{ik} - Feature frequency of term
α – Weight assigned to Title region
β - Weight assigned to Hyperlink region
μ – Weight assigned to body Region

In order to calculate feature frequency of term appearing in regions of web page, frequency of term in each region is considered. The more weightage is assigned term appearing in title region followed by hyperlink region and body region i.e. $\alpha > \beta > \mu \geq 1$

$$tf_{ik} = \alpha \text{ x } tf_{ik1} + \beta \text{ x } tf_{ik2} + \mu \text{ x } tf_{ik3}$$

While calculating feature frequency of term $tf_{ik,}$ multiply it by factor $\log_2 (M/ m_i)$ where M is the number of feature item appearing in document and m_i is the number of number of item appearing in the i^{th} region.

$$tf_{ik} = \alpha \text{ x } tf_{ik1} \text{ x} \log_2(M/ m_1) + \beta \text{ x } tf_{ik2} \text{ x } \log_2(M/ m_2) + \mu \text{ x } tf_{ik3} \text{ x } \log_2(M/ m_3) \qquad \text{Eq. (1)}$$

Where $M = m_1 + m_2 + m_3$

The term idf represents inverse document frequency and is given by Eq (2)

$$idf_k = \log\left(\frac{N}{n_k}\right) \qquad \text{Eq. (2)}$$

Where
N= total number of documents in the collection C
n_k = total number of documents in the C that contain term k

The weight of a term is the product of its feature frequency and inverse document frequency. This is given by Eq. (3)

$$w_{ik} = \frac{tf_{ik} \log(N / n_k)}{\sqrt{\sum_{k=1}^{t} (tf_{ik})^2 [\log(N / n_k)]^2}} \qquad \text{Eq. (3)}$$

The Similarity between document D_i and query q is defined as dot product of the document and query vectors which is equal to the cosine angle between document and query.

Let $w_{11}, w_{12}, \dots, w_{1t}$ represents weights of term appearing in document D_1. Let $w_{21}, w_{22}, \dots, w_{2t}$ represents weights of term appearing in query $q_{2.}$ The similarity is between document and query is calculated by Eq. (4). The Eq. (4) is normalized because document and query are of different length.

$$D_1 = w_{11}, w_{12,} ..., w_{1t}$$

$$q_2 = w_{21}, w_{22,} ..., w_{2t}$$

$$sim(D_1, q_2) = \sum_{i=1}^{t} w_{1i} * w_{2i} \quad \text{unnormalized}$$

$$sim(D_1, q_2) = \frac{\sum_{i=1}^{t} w_{1i} * w_{2i}}{\sqrt{\sum_{i=1}^{t} (w_{1i})^2 * \sum_{i=1}^{t} (w_{2i})^2}} \quad \text{cosine normalized Eq (4)}$$

4. DATA STRUCTURE

In N-level vector space model**,** the document is divided in three layers namely Title region, hyperlink region and body region. In order to record the presence and count the frequency the terms in the three regions, the sparse matrix is used

Because of the sparse matrix representation of document, storage space required is more. The sparse matrix often has large in size which has only a small number of nonzero elements. For sparse matrix having so many zero elements, therefore there need of appropriate data structure and methods to save storage and the number of operations. In order to solve the problem, compressed row storage format is used

4.1 Compressed Row Storage Format

The compressed row storage (CRS) format [4] puts the subsequent non-zeros of the matrix rows in contiguous memory locations. For non-symmetric sparse matrix **A**, three vectors are created

One for floating point numbers (val) and the other two for integers (col_ind, row_ptr). The val vector stores the values of the nonzero elements of the matrix **A** as they are traversed in a row-wise fashion. The col_ind vector stores the column indexes of the elements in the val vector.

if val(k)=a_{ij} , then Col_ind(k) = j . The row_ptr vector stores the locations in the val vector that start a row; that is, if val(k)=a_{ij} , then row_ptr(i) ≤ row_ptr(i+1).

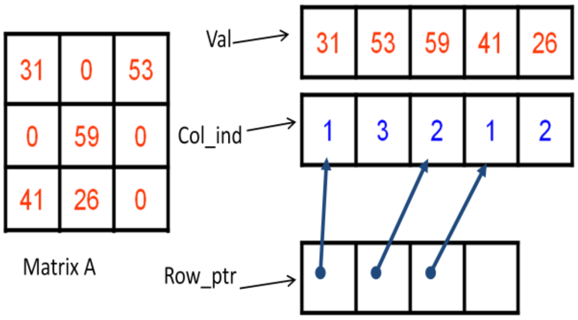

Fig 1 Compressed Row Storage Format

Val :- The non-zero elements are mapped into the val array.

col_ind:-Element i of the integer array, col_ind is the number of the column that contains the i-th element in the values array.

row_ptr:- Element j of the integer array, row_ptr gives the index of the element in the values array that is first non-zero element in a row j.

The length of the val and col_ind arrays is equal to the number of non-zero elements in the matrix. As the row_ptr array gives the location of the first non-zero element within a row, and the non-zero elements are stored consecutively.

While implementing N-level vector space model, stemming is used. A stemming is a process of linguistic normalization, in which the variant forms of a word are reduced to a root form

4.2 Stemming

The words that appear in documents and in queries often have many morphological variants. Thus pairs of terms such as "computing" and "computation" will not be recognized as equivalent

In most cases, morphological variants of words have similar semantic interpretations and can be considered as equivalent. For this reason a Porter's stemming algorithm [11] is used. Stemming algorithm reduces a word to its stem or root form. Thus the key terms of a query or document are represented by stems rather than by the original words. It reduces the dictionary size i.e. The number of distinct terms needed for representing a set of documents. A smaller dictionary size results in a saving of storage space and processing time. The algorithm for N-level vector space model is given below

N-level Vector Space model Algorithm

```
Assumption:
Let D = {   D_1, D_2, D_3, . . . . . . . . . . D_n }  be the Document set
      denote the  documents  in N-level vector space model
    tf – denote frequency of term in document
    idf – denote inverse document frequency of term

1.   Represent the document in vector space
   For each D_i in the Document set.
       Document is divided in the following regions.
            Title Region
            Hyperlinks region
            Body region
       Remove the Stop word
       Find all tokens in title region
       Find all token in the body region
       Find all the Hyperlinks and Anchors in the document
       Apply Porter's Stemming algorithm[11]  to remove
        morphological variants of all terms
   End for
2.   Prepare Dictionary of terms
3.   The documents are represented in N- layer vector space
     using sparse matrix.
4.    Calculate feature frequency tf using Eq. 1
5.    Calculate  Inverse document  frequency  idf  using  Eq. 2
6.    Calculate  weight of term  tf * idf  i.e. weight of term
     Eq. 3
7.    Calculate  similarity between document  and query using
     sim ( D, q ) using Eq. 4
```

5. RESULT

N-level vector space approach is compared with classic vector space approach. For this purpose, Health-services and career-services web-data of University of Waterloo is used. In the experiment, the time required for create dictionary and indexing the web pages using vector space approach and N-level vector space approach is calculated. The result shows that the N-level vector space takes less time as compare to classic vector space model. The table 1 show time required to create dictionary and indexing with two dataset

Table 1 Time required for creating dictionary and indexing

Retrieval Model Data set	Vector space model (minute)	N-level space model (minute)
Health-service document set	6.32	3.12
Career-service document set	23.33	17.52

The precision and recall rate is used to evaluate the performance of vector space model and N-level vector space model.

Table 2 Precision and recall Rate for VSM and N-level VSM

Query	VSM Precision	VSM Recall	N-level VSM Precision	N-level VSM Recall
skin cancer	0.45	0.60	0.36	0.50
Flu	0.14	0.33	0.20	0.33
Chicken Pox	0.18	0.25	0.18	0.50
FED	0.18	0.25	0.18	0.50
Tb	0.09	0.50	0.13	1.00
Warts	0.05	1.00	0.07	1.00
Skin	0.36	0.63	0.36	0.88
Disease	0.23	0.40	0.32	0.43
Health Services	0.5	0.55	0.50	0.82
HIV AIDS	0.27	0.50	0.23	0.80
Acne	0.14	0.33	0.25	0.67
Lactose intolerance	0.09	0.50	0.13	0.50
Surgeries	0.32	0.57	0.36	0.63
Enzyme	0.23	0.60	0.27	0.67
Average	**0.23**	**0.50**	**0.25**	**0.66**

The table 2 shows the precision and recall for each query with vector space approach and N-level vector space approach is calculated and later average precision and average recall rate is calculated

Precision indicate proportion of items retrieved that are relevant and recall indicates proportion of relevant items that are retrieved.

$$Precision = \frac{\text{Number of retrieved relevant document}}{\text{Total number of retrieved document}} \quad \text{Eq. (5)}$$

$$Recall = \frac{\text{Number of retrieved relevant document}}{\text{Total number of releted retrieved document}} \quad \text{Eq. (6)}$$

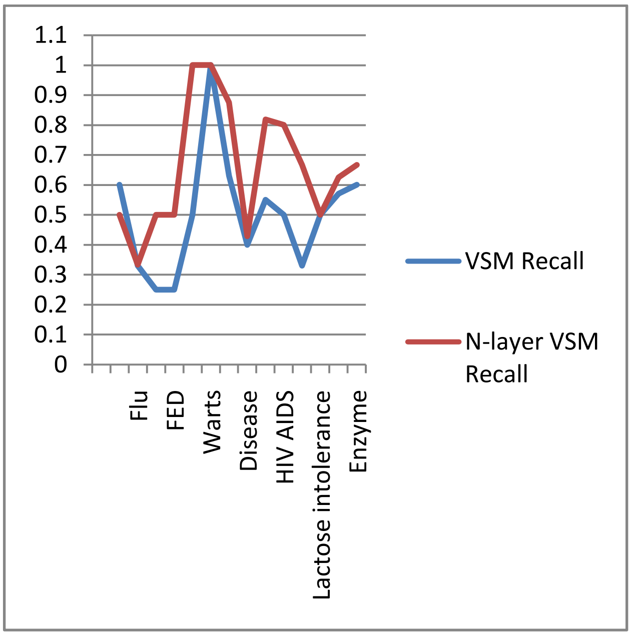

Fig 2 Recall comparison of VSM and N-level VSM

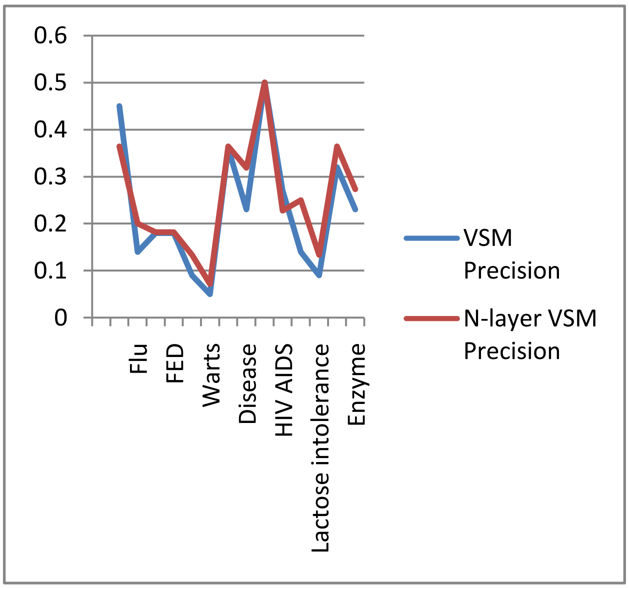

Fig 3 Precision Comparison for VSM and N-level VSM

In vector space model, recall rate varies from 0.25 to 1 and in N-level vector space model, recall rate varies from 0.33 to 1. The average recall rate in vector space model is 0.5 whereas in N-level average recall rate is 0.66.

In vector space model, precision varies from 0.05 to 0.50 and in N-level vector space model, precision varies from 0.07 to 0.50. The average precision vector space model is 0.23 whereas in N-level average precision is 0.25.

6. CONCLUSION

The area of web information retrieval has been researched extensively due to its current prime usage. The work one can be divided in two parts. Implementation of vector space model and N-level vector space model for which the standard tested data set like Health-services and career-services web-data of University of Waterloo is used.

The second part uses typically sparse Matrix. This work suggests compression of sparse matrix using row compression storage approach. Compressed row storage (CRS) format puts the subsequent non-zeros of the matrix rows in contiguous memory locations. The memory storage savings with this approach is significant. In a scenario, when there are m documents with n terms in matrix, for storage need m x n matrix element. Using row compression storage, the storage requirement reduces to 2z + m +1 storage space, where z enumerates the total number of non-zeros in the matrix and m denote the number of rows.

N-level vector space model requires less time for creating dictionary and indexing the documents as compare to classic vector space model. The experiment shows for health service dataset, vector space model requires 6.32 min whereas N-level vector space model requires 3.12 min for indexing of documents from dataset. Similarly with Career-service dataset, vector space model requires 23.33 min whereas N-level vector space model requires 17.52 min for indexing of documents from dataset.

The average recall rate in vector space model is 0.5 whereas in N-level average recall rate is 0.66. N-level vector space approach, recall rate is better as compare to classic vector space approach. Similarly the average precision in vector space model is 0.23 whereas in N-level average precision is 0.25. N-level vector space approach shows very small improvement when it is compared to classic vector space approach.

7. REFERENCES

[1] Srinath Sriniwas, P.C. Bhatt (2002) “Introduction to Web Information Retrieval: A User Perspective” Resonance June 2002 Resonance, June 2002 Page 27-38

[2] P. Ravikumar, Ashutosh kumar singh (2010) “Web Structure Mining: Exploring Hyperlinks and Algorithms for information Retrieval” American Journal of Applied Science 7(6) 2010 Page 840-845

[3] Anwar A. Alhenshiri “ Web Information Retrieval and Search Engine Techniques” Al-Satil Journal Page 55-81

[4] Nazli Goharian, Ankit Jain, Qian Sun “Comparative Analysis of Sparse Matrix Algorithms for Information Retrieval” Systemics Cybernetics and informatics volume 1, page 39- 46

[5] Changxia Hu, Xiaoxing Liu, Weiying Jin “Research on the Web Information Retrieval Model Based on Metadata and Query Expansion” 978-1-4244-4900-2/09 2009 IEEE.

[6] Mehran Sahami, Vibhu Mittal, Shumeet Baluja, Henry Rowley. “The Happy Searcher: Challenges in Web Information Retrieval” Google Inc. 1600 Amphitheatre Parkway, Mountain View, CA 94043

[7] Ricardo Baeza-Yate “Information retrieval in the Web: beyond current search engines” International Journal of Approximate Reasoning 34 (2003) 97–104

[8] Joon Ho Lee, “Properties of Extended Boolean models in information Retrieval” Korea research and development center, koera institute of science and technology

[9] Weiqun Luo, Chungui Liu, Zhiwei Liu, Conghua Wang (2010) “On N-layer Vector Space model-based Web Information Retrieval” 978-I-4244-3709-2/10 IEEE

[10] The Porter stemming algorithm, available at http://snowball.tartarus.org/algorithms/porter/ stemmer.html.