Keyword-Based Information Retrieval System b Utilizing Vector Space Model (VSM)

Theingi Shwe, Khin Mar Myo
University of Computer Studies, Mawlamyine
geolugi@gmail.com, kmmyo09@gmail.com

Abstract

The majority of today's Information Retrieval tasks are based on two main processes: indexing and searching. Indexes and retrieval are at the core of every modern Information Retrieval (IR) System. Many IR systems are based on vector space model (VSM). This paper is focused attention on the VSM of searching and retrieving on "Natural Disaster" information. In this system, index table is used for data structure and VSM also applied to retrieve information which is user's desirable information. Cosine similarity method is used to compute the similarity between query vector and document vector. HTML parser is used in our system for parsing the HTML pages. And we also use the term frequency and inverse document frequency (tf-idf) weighting scheme for the document indexing. Query is formulated by using terms and then this query is evaluated by processing the index to retrieve all stored information by an appropriate combination of query words.

1. Introduction

There is an increasing trend of storing web documents and literature as free text on the World Wide Web (WWW). As a result, the accurate retrieval of these documents become increasingly important.

IR can be defined broadly as the study of how to determine and retrieve from a corpus of stored information which is relevant to particular information needs. IR system has found many applications due to the abundance of text information. The heart of an IR system is its retrieval model. The model is used to capture the meaning of documents and queries.

There are three classic models in IR. They are(i) Boolean Retrieval Model where document and query are represented as sets of index terms. This model is said that set theoretic, (ii) Vector Retrieval Model where document and query are represented as vectors in a dimensional space. This model is said that algebraic, and (iii) Probabilistic Retrieval Model where framework for document and query representations is based on probability theory. This model is said that probabilistic.

Many IR systems are based on VSM that represents a document as a vector of index terms. There are many vector space models. These models are stem-based VSM, concept-based VSM and phrase-based VSM. Stem-based VSM represents a document as a vector of terms in VSM. The basis of the vector space corresponds to distinct terms in a document collection [5].

Indexing is used to facilitate the retrieval of such documents. VSM [2] is widely used to index documents. Under VSM, a document is represented by a vector of terms *document vector*). The cosine of the angle between two document vectors indicates the similarity between the corresponding documents. A smaller angle corresponds to a larger cosine value and indicates higher document similarity. A query, which describes the information need, is encoded as a vector as well. Retrieval of documents that satisfy the information need is achieved by finding the documents most similar to the query, or equivalently, the document vectors closest to the query vector.

This paper describes how to effectively use the VSM in the implementation searching and retrieving offline documents for domain application of natural disaster.

The rest of the paper is organized as follows: related work is described in the next section and background theory is described in section 3. Architecture of the system and step by step process is explained in section 4. Next, section 5 explain the implementation of the system and we consider the performance of the system. Finally, we conclude this paper and provide the future work.

2. Related Work

James and Wesley [4] proposed multilist files to retrieve information which satisfy Boolean query. In mutlilist files, the next record on a list is located using a pointer which is stored on the record precedes it on the list. While processing the multilist files, one or more pointers associated with the attributes may

point at records preceding the ones just read. As more space, the costs of storage are also higher.

The main limitation of the Boolean retrieval model is its incapability to rank the result and to match documents that do not contain all the keywords of the query. In addition, more complex requests become very difficult to formulate. The Vector space model address these issues, by supporting non-binary weight i.e. real number in (0,1) , both for documents and queries, and producing continuous similarity measures in (0,1). The similarity measure is derived from the geometrical relationship of vectors in the t-dimensional space of document/query used both on the Web and for classical text retrieval. And cosine similarity measurement is used to search the similarity between terms of query and documents [2].

According to literature review, VSM technique is very useful relevant retrieve. Therefore, this paper takes the advantages of VSM in the implementation of searching and retrieving "Natural Disaster" information from the collection of document.

3. Background Theory

This section reviews on some background theory concerning with our system.

3.1 Text Retrieval for IR

IR is concerned with the organization and retrieval of information from a large number of text-based documents. Text retrieval methods fall into two categories; they are generally view retrieval problem as a document selection problem and as a document ranking problem [1]. In document selection method, the query is regarded as specifying constraints for selecting relevant documents. A typical method of this category is the Boolean retrieval model, in which a document is represented by a set of keywords and a user provides a Boolean expression of keywords. Document ranking methods use the query to rank all document in the order of relevance. Most modern IR systems present a ranked list of document in response to user's keyword query. The most popular approach in this method is VSM.

3.2 Vector Space Model

The vector space model procedure can be divided in to three stages. The first stage is the document indexing where content bearing terms are extracted from the document text. The second stage is the weighting of the indexed terms to enhance retrieval of document relevant to the user. The last stage ranks the document with respect to the query according to a similarity measure [1].

The term frequency (tf) for term in document can be formulated as:

$$tf(i,j)=\begin{cases}0 & \text{if freq } (i,j) = 0 \\ 1+\log(1+\log(\text{freq } (i,j))) & \text{otherwise}\end{cases} \quad (1)$$

Where freq (i,j) is the number of occurrences of term in the document, tf (i,j) is the association of term with respect to the given document.

In equation (2), idf (i) is inverse document frequency where d is number of document in document in document collection and d_t is the set of document containing term.

$$idf(i) = \log \frac{1+|d|}{|d_i|} \quad (2)$$

A resonable measure of term importance many then be obtained by using the product of term frequency and the inverse document frequency: A third a term-weightig factor, in addition to the term frequency and inverse document frequency is useful in web-based IR system with widely varying vector lengths.

$$w_{ij} = tf\_idf(i,j) = tf(i,j)*idf(i) \quad (3)$$

For each document and query, compute all vector lengths (zero terms are ignored).

$$|D_i| = \sqrt{\sum_i w_{i,j}^2} \quad (4)$$

$$|Q| = \sqrt{\sum_i w_{Q,j}^2} \quad (5)$$

The vector product measures the number of terms that are assigned to query Q and document D. Dot product formula is defined as:

$$Q * D = \sum_i w_{Q,j} w_{i,j} \quad (6)$$

The vector similarity function is the well known cosine vector similarity formula in web-based IR system would be used as:

$$Sim(Q, D_i) = \frac{\sum_i w_{Q,j} w_{i,j}}{\sqrt{\sum_j w_{Q,j}^2} \sqrt{\sum_i w_{i,j}^2}} \quad (7)$$

4. Architecture of the System

The overview of the system is described in Figure 1. This system is intended to provide the relevant information between the user's desirable information and collection of web pages. There are two processing phases in this system: (1) preprocessing steps for Web-based IR system and (2) retrieving process of the system.

4.1 Preprocessing Step for the System

In the processing steps, firstly web pages parsing is done. In the process of web-page parsing, web advertiser can advertise their advertising web pages into the web-based IR system. After the page registration is completed, web-based IR system parses the registered pages by removing HTML tags and other attributes, CSS properties. Texts inside some of the tags have to be totally removed such as (SCRIPT) etc… Some of the tags have to be removed not the text inside the tags, for example <Title> and <Hn>. After parsing the HTML source code, text is extracted between the tags. In the tokenization process , extracted text is tokenized, if the stop words are contained, they are first removed and then remaining terms are converted to lowercase. The document can only be represented by content bearing words. This indexing based on term frequency. The index table is constructed with the content bearing words by calculating of each terms for all documents. After that, index table is stored into the database at dynamically. Query is also worked as the process flow of the document.

4.2. Retrieving Stage of the System

In retrieving, the user input the query. The query process of VSM that the user's input query is first tokenized into individual terms and stop words are removed. And then, these terms are matched with terms in index table whether these terms are in the index table or not. Then the calculation of weight of each term for the query and documents based on tf-idf weighting scheme. The cosine similarity method is used to measure how similar document is to a query. These similarity values are sorted by decreasing order. So the top document is the most relevant to user's desirable information because the similarity result is the highest order. As a result, the result page with link is retrieved which is relevance to user's query and collection of web pages.

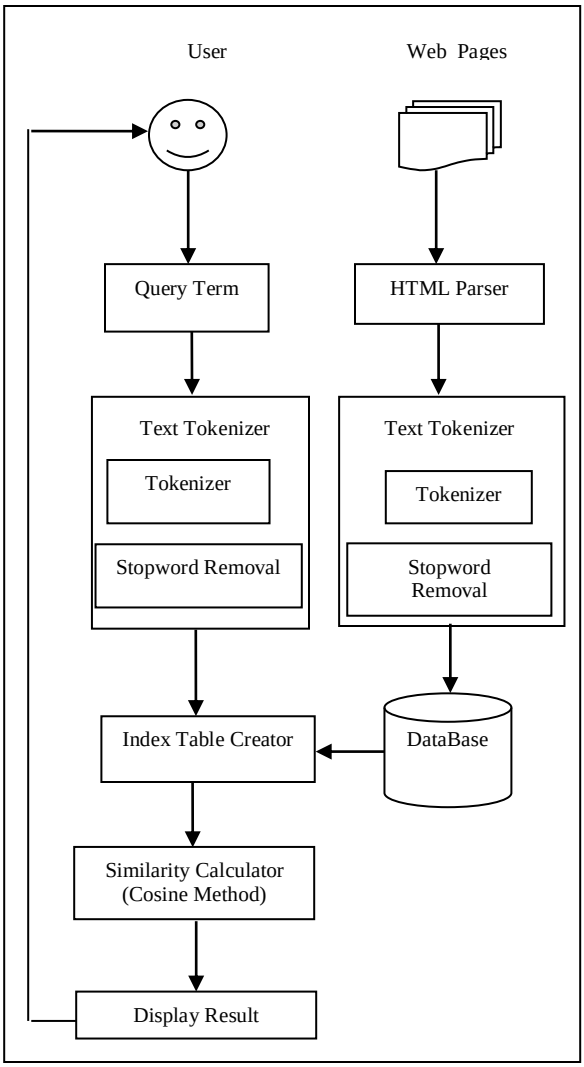

Figure 1. Architecture of the System

5. Performance Measure and Results

Experiment is conducted based on the collection of Natural Disaster Documents to measure the strength of VSM .This section describes not only the detail of experiments but also the results from this experiments. This system uses the collection HTML pages for experiment.

There are two main phases to conduct the experiment: (i) Preprocessing phase and (ii) Retrieving Relevant Document phase.

5.1 Preprocessing Phase

As describe in section 4.1 the existing document (HTML web page) is needed to preprocess with the following steps.

(i) Parsing HTML Page

Before the calculation of weighting and similarity, at first HTML page is parsed. For example, when the document (D1) is parsed this system.

<html>
<head>
<title></title>
</head>
<body>
<h1> Tropical Cyclone Nagis, which struck Myanmar in May 2008 </h1>
</body> </html>

After passing the HTML page, is to extract the content between HTML tags. In this system, the contents between <title>,<h>, <p> and <meta> tags are extracted. According to above example, the following content is obtained.

" Tropical Cyclone Nargis, which struck Myanmar in May 2008"

(ii) Tokenizing the Web Content

The extracted content terms are tokenized and removed the stop words are as follow:

"Tropical Cyclone Nagis struck Myanmar May 2008"

These words are changed into lowercase letter.

"tropical cyclone nagis struck myanmar may 2008"

5.2 Retrieving Phase

Let the following is the contents from another web documents from documents D1 to D5

D1*: Typhoon Chan-hom, which struck Philippines in May 2009*

D2*: Tropical Cyclone Bijli, which struck Bangladesh in April 2009*

D3*: Flood in China as on June 2008*

D4*: Earthquake in Sichuan Province, China as on May 2008*

D5*: Tropical Cyclone Nagis, which struck Myanmar in May 2008.*

After parsing the phase 5.1, the above contents are converted into the following work tokens as follow.

"typhoon chan-hom, philippines may 2009 "

"*tropical cyclone bijli bangladesh april 2009 "*

"flood china june 2008 "

"earthquake sichuan province china may 2008"

"tropical cyclone nargis myanmar may 2008"

Suppose the query term is "Cyclone in 2008".

When user input the query from the input box, query is tokenized and removed the stop words.

The query terms are："cyclone 2008"

Calculate the weight in indexing step. The index tables are constructed by analyzing the terms of all documents and query and find the frequency of each term in all documents and query. Table 1 is term frequency matrix showing the frequency of terms per document. Tf-idf is calculated by using equation 1 and 2 in section 3. By using equation 3, Table 2 is constructed.

VSM constructs the index tables is shown in Table 1 and Table 2 by analyzing the terms of all documents into words as in Table 1 and search the weight of each term in all documents, Table 2, does the same for the query. These tables are stored in database at dynamically.

Table 1. Frequent Count for Term

Terms	Q	D_1	D_2	D_3	D_4	D_5
april	0	0	1	0	0	0
bangladesh	0	0	1	0	0	0
bijli	0	0	1	0	0	0
chan-hom	0	1	0	0	0	0
cyclone	1	0	1	0	0	1
china	0	0	0	1	1	0
earthquake	0	0	0	0	1	0
floods	0	0	0	1	0	0
june	0	0	0	1	0	0
may	0	1	0	0	1	1
myanmar	0	0	0	0	0	1
nagis	0	0	0	0	0	1
philippines	0	1	0	0	0	0
province	0	0	0	0	1	0
sichuan	0	0	0	0	1	0
tropical	0	0	1	0	0	1
typhoon	0	1	0	0	0	0
2008	1	0	0	1	1	1
2009	0	1	1	0	0	0

Table 2. Weight for each Term

Terms	Q	D_1	D_2	D_3	D_4	D_5
april	0	0	0.778	0	0	0
bangladesh	0	0	0.778	0	0	0
bijli	0	0	0.778	0	0	0
chan-hom	0	0.778	0	0	0	0
cyclone	0.477	0	0.477	0	0	0.477
china	0	0	0	0.477	0.477	0
earthquake	0	0	0	0	0.778	0
floods	0	0	0	0.778	0	0
june	0	0	0	0.778	0	0
may	0	0.301	0	0	0.301	0.301
myanmar	0	0	0	0	0	0.778
nagis	0	0	0	0	0	0.778
philippines	0	0.778	0	0	0	0
province	0	0	0	0	0.778	0
sichuan	0	0	0	0	0.778	0
tropical	0	0	0.477	0	0	0.477
typhoon	0	0.778	0	0	0	0
2008	0.301	0	0	0.301	0.301	0.301
2009	0	0.477	0.477	0	0	0

5.3 Similarity Analysis

Cosine method is used to measure the similarity a document is to a query in this system. The similarity measure for the previous section 5.1 can be calculated as followed. For each document and query, compute all vector lengths by using equation 4 and 5.

$$|D_1| = \sqrt{0.778^2 + 0.301^2 + 0.778^2 + 0.778^2 + 0.477^2} = 1.461$$

$$|D_2| = \sqrt{0.778^2 + 0.778^2 + 0.778^2 + 0.477^2 + 0.477^2 + 0.477^2} = 1.581$$

$$|D_3| = \sqrt{0.477^2 + 0.778^2 + 0.778^2 + 0.301^2} = 1.124$$

$$|D_4| = \sqrt{0.477^2 + 0.778^2 + 0.301^2 + 0.778^2 + 0.778^2 + 0.301^2} = 1.492$$

$$|D_5| = \sqrt{0.477^2 + 0.301^2 + 0.778^2 + 0.778^2 + 0.477^2 + 0.301^2} = 1.359$$

$$|Q| = \sqrt{0.477^2 + 0.301^2} = 0.564$$

By using equation 6, compute all dot product.

$Q * D_1 = 0$

$Q * D_2 = 0.477 * 0.477 = 0.228$

$Q * D_3 = 0.301 * 0.301 = 0.091$

$Q * D_4 = 0.301 * 0.301 = 0.091$

$Q * D\ 5 = 0.477*0.477 + 0.301 * 0.301 = 0.319$

Calculate the similarity values by using equation 7.

$$\text{Cosine } \theta_{D1} = \frac{Q * D1}{|Q| * |D1|} = 0$$

$$\text{Cosine } \theta_{D2} = \frac{Q * D_2}{|Q| * |D_2|} \quad \frac{0.228}{0.564 * 1.581} = 0.150$$

$$\text{Cosine } \theta_{D3} = \frac{Q * D_3}{|Q| * |D_3|} = \frac{0.091}{0.564 * 1.124} = 0.084$$

$$\text{Cosine } \theta_{D4} \quad \frac{Q * D_4}{|Q| * |D_4|} = \frac{0.091}{0.564 * 1.492} = 0.063$$

$$\text{Cosine } \theta_{D5} \quad \frac{Q * D_5}{|Q| * |D_5|} = \frac{0.319}{0.564 * 1.359} = 0.416$$

In our system, the threshold value is set "zero" (th=0), because the main aim of system is to retrieve relevance document rather than exact document. Therefore the documents with the condition of similarity (sim) value is greater than zero (sim>th) are retrieved as relevance documents.

Among five documents in above example, four documents have been retrieved because they match with the above condition (sim>th). The similarity result is ordered by descending. The results are following.

Cosine $\theta_{D5} = 0.416$

Cosine $\theta_{D2} = 0.150$

Cosine $\theta_{D3} = 0.084$

Cosine $\theta_{D4} = 0.063$

Document D5 is the most relevant to user's desirable information because of the similarity result of D5 is the highest order. Then display the result to user. The displayed results are as follows:

D5: *Tropical Cyclone Nagis, which struck Myanmar in May 2008*

D2: *Tropical Cyclone Bijli, which struck Bangladesh in April 2009*

D3: *Flood in China as on June 2008*

D4: *Earthquake in Sichuan Province, China as on May 2008*

5.3 Performance Evaluation

In this section, the performance of our system is evaluated. We first compute the precision and recall for this system. To access the "accuracy" or "correctness" of the system, there are two measures of IR success, both based on the concept of relevance [to a given query or information need], are widely used: "precision" and "recall".

Precision: how many of the documents the system retrieved are correct (relevant to the query).

Recall : how many of the relevant documents in the collection the system managed to find.

Precision and recall can be defined with matrix tabulates in terms of:

	Relevant	Non-relevant
Retrieved	True positives(TP)	False positive (FP)
Not retrieved	False negative(FP)	True negative(TN)

Precision and recall would be formally defined as:

Precision (P): $P = \frac{TP}{TP + FP}$

Recall (R): $R = \frac{TP}{TP + FN}$

In order to evaluate the system, document collection with 150 documents are used, 28 of which are relevant for a given query. Query length one "cyclone" retrieves 10 documents, 8 of which are relevant: TP = 9, FP = 1, FN = 20.

$$P = \frac{TP}{TP + FP} = \frac{9}{9 + 2} = 0.83$$

$$R = \frac{TP}{TP + FN} = \frac{9}{9 + 20} = 0.3$$

Query length two retrieves TP = 15, FP = 12, FN = 16. Precision = 0.80, Recall = 0.43.

Query length three retrieves TP = 20, FP = 5, FN = 13. Precision = 0.75, Recall = 0.53.

Query length four retrieves TP = 16, FP = 9, FN = 12. Precision = 0.64, Recall = 0.57.

Precision and recall curve is described in table 3 for the performance evaluation of the system.

Table 3. Precision and Recall

Query length	One	Two	Three	Four
Recall	0.3	0.43	0.53	0.57
Precision	0.83	0.80	0.75	0.64

6. Conclusion and Future Work

This paper, has described the design and implementation of IR system based on VSM .To sum up, the goal of this paper is to provide a framework for building index table and operating of searching retrieving by VSM and similarity analysis. To realize and simulate the process of IR. This paper is mentioned two stages of IR system: Indexing and Retrieving stage. Indexing based on tf-idf weighting scheme. Similarity between a document and query is determined by the cosine method. The result is returned to user which is relevant to information need. But stemming and semantics meaning isn't considered in our system and input data is predefined.

For future work, stemming and other linguistic modules can be applied in building index table. This index table can be compressed if the index table takes less storage and will reduce I/O time.

Reference

[1] J. Han, M.Kamber, "Data Mining Concept and Techniques".

[2] G.Salton, C.Buckley, " Term-Weighting Approaches in Automatic Text Retrieval".

[3] M.S.Abual-Rub,R.Abdulah, N.A.A.Rashid "A Modified Vector Space Model for Protein Retrieval "

[4] W.J. Welch, J.W. Grahem, "Retrieval Using Ordered Lists in Inverted and Multilist Files", *Department of Computer Science, University of Waterloo, Ontario,Canada.*

[5] W.Mao, W.W.Chu,"Free-Text Medical Document Retrieval Via Phrase-based Vector Space Model".