Text Mining System Based on Vector Space Model

Chit Hnin Aye, Pearl
Computer University, Pathein
chithnin88@gmail.com, pearl417@gmail.com

Abstract

Evidently there is a tremendous proliferation in the amount of information found today on the largest shared information source, the World Wide Web. Information Retrieval System tries to save the users access time by classifying the documents and clustering the documents because users spend a lot of time to find documents or information from texts. Most text mining system refers to tasks user information retrieval method to pre-process text documents. Text mining system serves as a powerful technique to manage the text in large document collections. Our proposed system had been developed by applying the pre-processing steps of text mining system and the vector space model (VSM). The system uses term frequency-inverse document frequency formula to calculate the weight of terms and query. Cosine measure of similarity formula sorts the documents according to the degree of similarity values. This paper intends to an effective text mining process by using Vector Space Model. By using this system, user can use as a song search engine and can listen this song and free download and upload.

Keywords: Vector Space Model, Text Mining

1. Introduction

The process of finding relevant information on the web can be overwhelming. It involves retrieving and integrating information from web document. Therefore, information retrieval system should provide advanced searching facilities to retrieve information. Information Retrieval is the searching for documents, for information within documents and for metadata about documents, as well as that of searching relation database and the World Wide Web. The vector space model has been widely used in the traditional information retrieval (IR) field. In the vector space model, treats text representation of objects and queries as vector in a multi-dimensional space, the dimensions of which are the words used to represent the terms. Most search engines also use similarity measures based on this model to rank Web documents. The model creates a space in which both documents and queries are represented by vectors.

There has been Term weighting is an important aspect of modern text retrieval systems. Terms are words, phrases, or any other indexing units used to identify the contents of a text. Queries and terms are compared by comparing the vectors, using the cosine similarity measure. Finally, the result sort and rank the documents in descending order according to the similarity values.

The rest of this paper is organized as follows. In section 2, Related Works is presented. The background theory is described in section 3 and proposes system overview is explained in section 4. In section 5, the similarity measurement and the experimental results are shown in section 6. We conclude this system in section 7.

2. Related Works

Semantic space and the distributional hypothesis have been widely and successfully applied to different language related tasks, such as information retrieval [8], the relationships among words have been widely studied in fields of nature language processing, text mining and information retrieval, etc. One method is Latent Semantic Indexing (LSI) [2], which automatically discovers latent relationships among corpora through Singular Vector Decomposition. However, the method is time-consuming when applied to a large corpus [3]. Vector Space search technology can be used on any type information that can be represented in a structured fashion, so it will work equally well on text, images, cryptographic keys, or even DNA.

3. Background Theory

This paper based on term frequency and inverse document frequency and calculates cosine similarity using vector space model and document preprocessing using text mining system.

3.1. Web mining

Web mining is moving the World Wide Web toward a more useful environment in which user can quickly and easily find the information they need. It includes the discovery and analysis of data, documents and multimedia from the World Wide Web.

Web mining can be divided into three different types, which are Web usage mining, Web structure mining, and Web content mining [1]. Web usage

mining is the types of Web mining activity that involves the automatic discovery of user access patterns from one or more Web servers. Web structure mining is the process of using graph theory to analyze the node and connection structure of a web site [1].

3.2. Web Content Mining

Web content mining is the process to discover useful information from text, image, audio or video data in the web. Web content mining sometimes is called web text mining, because the text content is the most widely researched area. Web content mining is related but different from data mining and text mining. It is related to text mining because much of the web contents are texts. Text mining is one of the categories of the web mining. Most text mining tasks use information retrieval methods to pre-process text documents [5]. The term frequency-inverse document frequency (tf-idf) is a weight often used in information retrieval and text mining.

3.2.1. Text mining

Text mining takes unstructured textual information and examines it in an attempt to discover structure and implicit meanings buried within the text. Text mining, sometimes alternately referred to as text data mining, roughly equivalent to text analysis, refers generally to the process of deriving high-quality information from text. High-quality information is typically derived through the dividing of patterns and trends through means such as statistical pattern learning. Text mining usually involves the process of structuring the input text (usually parsing, along with the addition of some derived linguistic features and the removal of others, and subsequent insertion into a database), deriving patterns within the structured data, and finally evaluation and interpretation of the output [6].

3.3. Terms Frequency (tf) and Inverse Document Frequency weighting (idf)

The term frequency (tf) and inverse document frequency (idf) is a weight often used in information retrieval and text mining. This weight is a statistical measure used to evaluate how important a word is to a document in a collection or corpus [10].The importance increase proportionally to the number of times a word appears in the document but is offset by the frequency of the word in the corpus. The definition of term depends on the application. Typically terms are single words, keywords, or longer phrases. If the words are chosen to be the terms, the dimensionality of the vector is the numbers of words in the vocabulary [7].

3.3.1. Term Frequency (tf)

The term frequency in the given document is simply the number of times a given term appears in that document. This count is usually normalized to prevent a bias towards longer documents (which may have a higher term frequency regardless of the actual importance of that term in the document) to give a measure of the importance of the term t_i within the particular document d_o.

$$tf_{i,j} = \frac{n_{i,j}}{\sum_k n_{k,j}} \qquad (1)$$

Where $n_{i,j}$ is the number of occurrences of the considered term in document d_j, and the denominator is the sum of number of occurrences of all terms in document d_j.

3.3.2. Inverse Document Frequency (idf)

The inverse document frequency is a measure of the general importance of the term (obtained by dividing the number of all documents by the number of documents containing the term, and then taking the logarithm of that quotient).

$$idf_i = \log \frac{|D|}{|\{d_j : t_i \in d_j\}|} \qquad (2)$$

$|D|$: Total number of documents in the corpus

$|d_j : t_i \in d_j|$: Number of documents where the term t_i appears (that is $n_{i,j} \neq 0$) then,

$$W_{ij} = tf_{ij} * idf_j \qquad (3)$$

A high weight in term frequency and inverse document frequency (tf_idf) is researched by a high terms frequency (in the given document) and a low document frequency of the term in the whole collection of documents, the weights hence tend to filter out common terms. The term frequency and inverse document frequency (tf_idf) weighting scheme is often used in the vector space model together with cosine similarity to determine the similarity between two documents [10].

3.4. Vector Space Model (VSM)

The Vector Space Model recognizes that the use of binary weights is too limiting and proposes a framework in which partial matching is possible. This is accomplished by assigning non-binary weights to index terms in queries and in documents. These term weights are ultimately used to compute the degree of similarity between each documents stored in the system and the user query. By sorting the retrieved documents in decreasing order of this degree of similarity, the vector space model takes into consideration documents which match the query

terms only partially. The main result effect is that the ranked document answer set is a lot more precise (in the sense that it better matches the user information need) than the document answer ser retrieved by the Boolean model [11, 2].

Similar documents are expected to have similar relative term frequencies, this similarity among a set of documents or between a document and a query (often defined as a set of keywords) can be measured, based on similar relative term frequency occurrences [10].

3.5. Similarity Computation in Vector Space Model

The vector model proposed to evaluate the degree of similarity of the document d_j with regard to the query q as the correlation between the vectors $\vec{d_j}$ and $\vec{q}$. This correlation can be quantified, for instance, by the cosine of the angle between theses two vectors. That is,

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

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

$$Sim(d_j,q) = \frac{\vec{d}}{|\vec{d}_j} \quad (6)$$

where | and are the norms of the document and query vectors. The factor does not affect the ranking (i.e., the ordering of the documents) because it is same for all documents. The factor | provides normalization in the space of the documents.

$$Sim(d_j, q) = \frac{\sum_{i=1}^{t} w_{i,j} * w_{i,q}}{\sqrt{\sum_{i=1}^{t} w_{i,j}^2} * \sqrt{\sum_{j=1}^{t}}} \quad (7)$$

Since $w_{i,j} \geq 0$ and $w_{i,q} \geq 0$, Sim (q, d_j) varies from 0 to +1. Thus, instead of attempting to predict whether a document is relevant or not, the vector model ranks the documents to their degree of similarity to the query. A document might be retrieved even if it matches the query only partially. For instance, one can establish a threshold on Sim (q, d_j) and retrieve the documents with a degree of similarity above that threshold [2].

4. Proposed System Overview

In this system, there are four main components: (i) Text preprocessing, (ii) calculate the weights of documents and query (iii) calculate the cosine similarity between the documents and query. The overview of this system is shown in figure 1.

The purpose of this system is to perform the some elementary mathematical concepts shows their application in the development of the vector space model and text mining system. Whenever the user enters a word (e.g., heart), this system searches songs titles with the same this word. Then search results are ranked according to similarity of user query terms and also satisfied the user's needs.

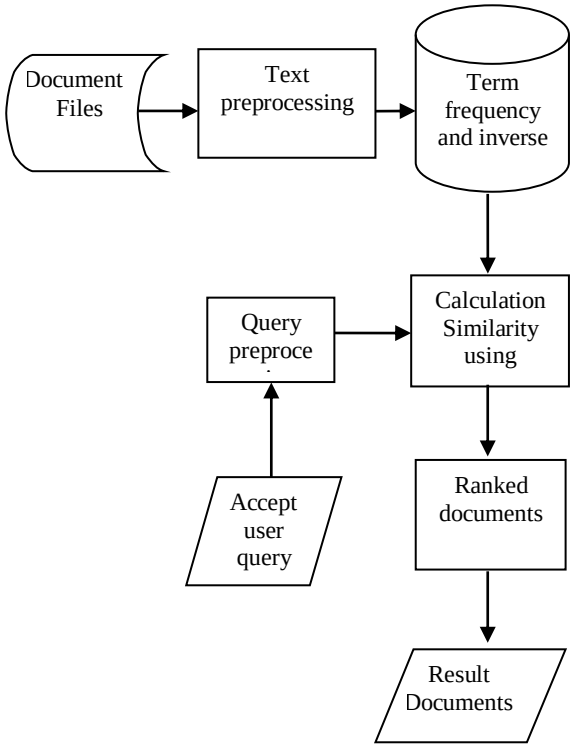

Figure 1. System Design

5. Similarity Measurement

Step 1: Text preprocessing takes four states, (1) Word (term) extraction, (2) Stop words removal, (3) Stemming, (4) Frequency counts and computing term frequency and inverse document frequency. In this system, construct an index of terms from the documents and determine the term counts tf_i for the query and each document D_j and then compute the document frequency d_i for each document. Since idfi = log (D/df_i) and D =5.Term frequency and inverse document frequency shown in table 1.

D1: “My love”
D2 : “Love of my life”
D3 : “Open your heart”
D4 : “I lay my love on you”
D5 : “Queen of my heart”

Figure 2. Song Documents

Q : My love

Figure 3.Input Query

Figure 2 shows the example song documents and the example query is “my love” (figure 3). We take the tf_i * idf_i products to compute the term weights from documents table.

Table 1. Example of Retrieval Terms from Song Documents

Terms	Tf						idf
	Q	D1	D2	D3	D4	D5	
My	1	1	1	0	1	1	0
Love	1	1	1	0	1	0	0.0969
Of	0	0	1	0	0	1	0.3979
Life	0	0	1	0	0	0	0.6989
Open	0	0	0	1	0	0	0.6989
Your	0	0	0	1	0	0	0.6989
Heart	0	0	0	1	0	1	0.3979
I	0	0	0	0	1	0	0.6989
Lay	0	0	0	0	1	0	0.6989
On	0	0	0	0	1	0	0.6989
Queen	0	0	0	0	0	1	0.6989
you	0	0	0	0	1	0	0.6989

Step 2: Now we treat weights as coordinates in the vector space, effectively representing documents and the query as vectors (Table 2). To find out the document vector is closer to the query vector, we resource to the similarity analysis introduced.

Step 3: We calculate the cosine similarity between the query and each term. Dot Product is the sum of the term counts for each document and the corresponding query term counts multiplied together. If a document doesn’t contain any relevant search terms from the query, the dot product will be zero. Finally, we sort and rank the documents in descending order according to the similarity values.

Ranking Documents
Rank 1 : Doc 1 = 1.0000
Rank 2 : Doc 2 = 0.1375
Rank 3 : Doc 4 = 0.0799
Output result -My love
- Love of my life
- I lay my love on you

Table 2. Weights Table

Terms	Weights = tf * idf					
	Q	D1	D2	D3	D4	D5
My	0	0	0	0	0	0
Love	0.0969	0.0969	0.0969	0	0.0969	0
Life	0	0	0.6989	0	0	0
Open	0	0	0	0.6989	0	0
Your	0	0	0	0.6989	0	0
Heart	0	0	0	0.3979	0	0.3979
I	0	0	0	0	0.6989	0
Lay	0	0	0	0	0.6989	0
Queen	0	0	0	0	0	0.6989
you	0	0	0	0	0.6989	0

5. Experimental Results

The user can search the songs at the home page. The search results are sorts according to the value of the similarity. The threshold similarity value is 0.5 in this system. We show that the example search result of the similarity value is grater than the threshold value as a search result in table 3.

Figure 4. Search Result Page

User can choose listen of like-minded and free download. The search result songs show in the figure 4.

Table 3. Example of Song Result (1)

Rank	Documents	Similarity value	Song name
1	Doc 1	1.0000	My love

Support another threshold similarity value is 0.1 then the search result is shown in table 4.

Table 4. Example of Song Result (2)

Rank	Documents	Similarity value	Song name
1	Doc 1	1.0000	My love
2	Doc 2	0.1375	Love of my life

By the above search result, the threshold value is most suitable as 0.5. If the threshold value is very small, less similarity documents also contain in search result. So, system needs more processing time and less performance.

6. Conclusion

A Text Ming System has been developed using vector space model. The vector space model of Information Retrieval is a powerful tool for constructing search machinery. The system provided an introduction to mathematical concepts required for understanding the vector model and showed the application of those concepts in the development of the model. In addition, this system provide easily search song's title, listen, and user can download within short time. Hence, users can free sign up (register). Furthermore, member can easily upload new songs.

7. References

[1] K.Aberer, *Information Retrieval and Data Mining Part 1 – Information Retrieval*, 2007/8.

[2] R. Ackerman, *Theory of information retrieval*, Florida State University, September 25.2003.

[3] S. Deerwester,., et al.: *Indexing by latent semantic analysis*. Journal of the American Society of Information Science 41(6), 391–407 (1990).

[4] M.Hearst, *Distinguishing between web data mining and information access,* August 16.

[5] M.Hearst, *What is text mining?* SIMS, UC Berkeley, October 17, 2003.

[6] Salton, A. Wong, and C. Yang. 1,975. A vector space model for automatic indexing. *Communications of the ACM*, 18:613–620.

[7] A.Scime, *Web Mining Applications and Techniques*, State University of New York College at Brockport, USA.

[8]V, *Basic Vector Space Search Engine Theory*, LA 2600, January 2, 2004.

[9] B.Yates Ricardo, *Morden Information Retrieval Book*, Pearson Education India.

[10]http://en.wikipedia.org/wiki/Vector_space_model

[11]http://www.ugcs.caltech.edu/~chandran/cs20/whatisvsm.html