Asian Journal of Information Technology 7 (1): 1-5, 2008
ISSN: 1682-3915

Inferring Document Similarity using the Fuzzy Measure

[1]K. Vivekanandan and [2]J. Suguna
[1]School of Management, University of Bharathiar, Coimbatore-46, Tamil Nadu, India
[2]Department of Computer Science, Vellalar College for Women, Erode-638 009, Tamil Nadu, India

Abstract: The aim of this study is to construct a system, capable of reading a collection of standard text file documents, performing semantic analysis on the documents and generating a similarity matrix used by the search engines, text corpus visualizations and a variety of other applications for filtering, sorting, clustering, retrieving and generally handling text. Methodologies used for this construction include statistical methods like Vector Space Model for document representation, latent semantic indexing method using singular value decomposition for dimension reduction and fuzzy measure for finding the similarity. The system could be implemented in Visual Basic integrated with MATLAB.

Key words: Fuzzy measure, latent semantic indexing, similarity matrix, singular value decomposition, vector space model

INTRODUCTION

The rapid progress of computer and network technologies make it easy to collect and store a large amount of unstructured or semi-structured texts such as books, magazine articles, research papers, product manuals, memorandums, e-mails and of course the web. They all contain textual information in the natural language form. This vast amount of textual information available today, is useless, unless, it can be effectively and efficiently handled. The automatic organization of this data has become an important research issue and a number of machine learning techniques have been proposed to enhance this process. This issue of identifying documents that might be of potential interest is increasingly being addressed by Data mining methods. In particular, a specialized form of Data mining, called Text mining has been used for identifying trends in text documents.

Document retrieval tasks are important when dealing with large databases of documents. Determining the similarity of documents is an important step for several document retrieval tasks such as document classification, categorization, sorting, filtering, information extraction and retrieval. All of these tasks require some notion of similarity. There are different methods available in the literature to compute similarities between documents (Rui and Wunsch, 2005). An alternative could arise from new representations of text documents, specifying new similarity models, or both. Hence, a similarity model should judge documents in terms of their similarity relationships to other documents in the corpus.

In all the information-retrieval systems, documents are grouped/filtered/sorted/retrieved on the basis of some chosen measure of similarity (Alexander, 2002). The most commonly used measure in document- processing is the Cosine Measure (Jiawei and Micheline, 2003). As an alternative to cosine measure method, Zadeh's Min-Max operations (Egghe, 2004) (one of the fuzzy set operations used in fuzzy information retrieval) can be used to construct the similarity matrix.

A fuzzy set is a generalization of the classical or crisp set with the range of [0, 1]. An object may only partially belong to a fuzzy set. The more the object belongs to the fuzzy set, the higher the degree of membership. Fuzzy Information Retrieval utilizes fuzzy sets to represent documents, membership degrees for query term relevance, fuzzy logical operators to define queries and fuzzy compatibility measures to assess the retrieval status value of a document. The measures provide an intuitive measurement of similarity and are also independent of the scale of the fuzzy sets (Egghe, 2004). Since fuzzy measures are robust against noise, distortions and resemble human reasoning, they result often in intuitively understandable linguistic rules. This study proposes fuzzy approach to measure closeness or similarity between the documents.

VECTOR SPACE MODEL

The representation and organization of the information items should be user -friendly. Hence, it is found that the Vector Space Model (VSM) (Jiawei and

Corresponding Author: K. Vivekanandan, School of Management, University of Bharathiar, Coimbatore-46, Tamil Nadu, India

Micheline, 2003) representation shows improvement over rudimentary methods in information retrieval which is described in this study.

Creating the term-document matrix: A database comprising a collection of textual documents is to be preprocessed by the following sequence of steps.

- Creating a list of all the words which appear in the documents.
- Removing words void of semantic content such as "and", "the", "of", etc. using the created list of stop words.
- Perform stemming using Porter Stemming Algorithm which is a process for removing the commoner morphological and inflectional endings from words in English (Porter, 1980).
- Further trimming the list by removing words which appear in only one document.

The remaining words are numbered as the terms, from 1 to m. Then create a m*n Term Document Matrix (TDM) (Salton *et al.*, 1975).

$$A = [a_{ij}]_{m \times n}$$

where i = 1,2,...,m (terms)

and j = 1,2,....,n (documents)

TERM WEIGHTING SCHEMES

The performance of the vector space model depends on the term weighting schemes, ie., the functions that determine the components of the vectors (Erica and Tamara, 1999). Proper term weighting can greatly improve the performance of the vector space model. A weighting scheme is composed of three different types of term weighting: Local, global and normalization (Salton *et al.*, 1975). The term weight is given by

$$a_{ij} = w_{ij} = t_{ij} * g_i * d_j \quad (1)$$

Where t_{ij} is the local weight for term i in document j, g_i is the global weight for term i and d_j is the normalization factor for document j.

Local weights are functions of how many times each term appear in a document, global weights are functions of how many times each term appears in the entire collection and the normalization factor compensates for discrepancies in the length of the documents.

Local weighting formulae perform well if they work on the principle that the terms with higher within-document frequency are more pertinent to that document. Global weighting tries to give a "discrimination value" to each term. Many schemes are based on the idea that the less frequently a term appears in the whole collection, the more discriminating it is. The third component of the weighting scheme is the normalization factor which is used to correct discrepancies in document lengths.

There are so many local, global and normalization weighting schemes available in the literature (Buckley, 1993). Among them, it has been found that the "log * global frequency inverse document frequency *cosine normalization" term weight scheme performs well and using this scheme the TDM is constructed.

DIMENSIONALITY REDUCTION

Since the number of terms and the number of documents are usually quite large, the high dimensionality of the term-document matrix leads to very sparse vectors and the problem of inefficient computation increases the difficulty in detecting and exploiting the relationships with terms. To overcome these problems, a Latent Semantic Indexing (LSI) method using Singular Value Decomposition (SVD) (Jiawei and Micheline, 2003) is used which effectively reduces the size of the term- document matrix for analysis.

Latent Semantic Indexing is an information retrieval method that organizes information into a semantic structure that takes advantage of some of the implicit higher-order associations of words with text objects. Through the pattern of co-occurrences of words, LSI is able to infer the structure of relationships between the documents and words. ie., LSI takes care of Polysemy (words having multiple meaning) and Synonymy (multiple words having the same meaning) problems that exist in efficient information retrieval. Documents which contain synonyms are closer in LSI space than in original space; documents which contain polysemy in different context are more farther in LSI space than in original space. The SVD algorithm preserves as much information as possible about the relative distances between the document vectors while collapsing them down into a much smaller set of dimensions. In this collapse information is lost and content words are superimposed on one another. Information loss sounds bad but here it is a blessing. What one loses is noise from the original term-document matrix revealing similarities that were latent in the document collection. Similar things become more similar while dissimilar things remain distinct. This reductive

mapping is what gives LSI its seemingly intelligent behavior capable of correlating semantically related terms.

The SVD method decomposes the original matrix A into three new matrices namely U, S and V such that But, our aim is to reduce A. (i.e.) obtaining an approximation of the original matrix. This is done by truncating the three matrices U, S, V. Essentially; we keep the first k columns of U, the first k columns of V and the first k rows and columns of S; (i.e.), the first k singular values. (The choice of k is done by "seat of the pants "). How many k singular values or dimensions to be kept are to be done more or less arbitrarily or must be determined experimentally since each collection is different (Harman, 1992). This removes noisy dimensions and exposes the effect of the largest k singular values on the original data.

FUZZY SIMILARITY MEASURE

Fuzzy vectors are one-dimensional array of membership values. Formally, a vector $a = (a_1, a_2, a_3,...a_n)$, is called a fuzzy vector if for any element, $0 \le w_i \le 1$ for I = 1,2,...n. In the obtained Term-document matrix, each column represents documents in vector representation. That is, $D_1 = (d_{i1})$ I = 1,2,...n, $D_2 = (d_{i2})$ I = 1,2,...n, $D_n = (d_{in})$, i = 1,2,...n. Here I = 1,2,...n symbolize n keywords and the numbers d_{i1}, d_{i2},...d_{in} represent the weight of keyword 'i' in documents D_1,D_2,...D_n, respectively. It is found that all d_{i1}, d_{i2},...$d_{in} \in [0,1]$. Hence, denoting the set of the n keywords by Ω (i.e.,) $\Omega = \{1,2,...n\}$, each document can be interpreted as a fuzzy subset of Ω as follows for D_1(and similarly for all D_i's): The membership function of D_1 is defined as:

$$\phi_{D_1} : \Omega \to [0,1]$$
$$i \to \phi_{D_1}(i) = d_{i1} \in \Omega$$

To compare 2 document vectors D and D' as above, it is clear that we have to study how close D and D' are. ie. we have to look for "common values" in each coordinate $I \in \Omega$, hence $\cap$ and $\cup$ should be used (Zadeh's min-max operations). In other words, comparing two documents is really looking for how much they are alike in each keyword i and hence $\cap$ and $\cup$ have to be used; making products of the weights of each keyword (as in cosine measure) is of no use here. This shows the fuzzy set applications of Zadeh's min-max operations. Thus comparing documents as vectors is an indexing action and this requires Zadeh's min-max operations (Egghe, 2004).

RESULTS AND DISCUSSION

A collection consisting of text documents are taken for processing (ftp://ftp.cs.cornell.edu/pub/smart). The following document indexing approaches are used: stopwords are removed; stemming is performed; unique terms are ignored. All this is done to reduce computational overhead (Fig. 1). Now, a term-document matrix (consists of only the content words) is constructed

Documents	No. of words	No. of words after preprocessing
50	4.37	37

Fig. 1: Document collection

	1	2	3	4	5	6	7	8
1	0.4898	0	0	0	0	0	0	0
2	0	0	0.4898	0	0	0	0	0
3	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0.4026	0
5	0.4898	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0
7	0.5192	0.4662	0.5192	0.4298	0.5192	0.8543	0.7274	0.7274
8	0	0	0	0	0	0	0	0
9	0	0	0	0.4026	0	0	0	0
10	0	0	0	0	0	0	0	0
11	0	0	0	0.4026	0.4898	0	0	0
12	0	0	0	0	0	0	0	0
13	0	0	0.4898	0	0	0	0	0
14	0	0	0	0	0	0	0	0
15	0.5007	0.4496	0.5007	0.4116	0.5007	0.5197	0	0
16	0	0	0	0	0	0	0.6862	0.6862
17	0	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0	0

37*50

Fig. 2: Term-document matrix

S-Matrix

	1	2
1	5.3449	0
2	0	1.8792

2*2

U-Matrix

	1	2
1	-0.046399	0.028557
2	-0.075394	0.05948
3	-0.033148	0.029478
4	-0.026482	-0084192
5	-0.035088	0.027179
6	-0.040377	0.035165
7	-0.7487	-0.036007
8	-0.023872	0.023194
9	-0.030382	0.041321
10	-0.039603	0.036562
11	-0.030553	0.025203
12	-0.016082	0.0098129
13	-0.042674	0.046754
14	-0.025247	0.015671
15	-0.63199	0.47125
16	-0.071561	-0.78318
17	-0.017284	0.026967
18	0.025118	0.023775

37*2

V-Matrix

	1	2
1	-0.1394	0.040606
2	-0.12506	0.043527
3	-0.14275	0.053767
4	-0.11803	0.043182
5	-0.13748	0.040734
6	-0.18112	-0.033364
7	-0.11307	-0.44339
8	-0.11100	-0.42535
9	-0.11817	0.025502
10	-0.13693	0.0043544
11	-0.15397	0.034576
12	-0.12386	0.033137
13	-0.18291	0.036166
14	-0.13786	0.041805
15	-0.11108	-0.42535
16	-0.15395	0.035861
17	-0.15924	0.047672
18	-0.10534	0.046059
19	-0.10171	0.050329

50*2

Fig. 3: Reduced matrices using SVD

	1	2	3	4	5	6	7	8
1	1	0.82532	0.83833	0.75757	0.97929	0.46061	0.12435	0.12479
2	0.82532	1	0.71852	0.90986	0.84277	0.38015	0.12235	0.12273
3	0.83833	0.71852	1	0.64537	0.82059	0.41488	0.10118	0.10088
4	0.75757	0.90986	0.64537	1	0.7736	0.34896	0.12449	0.12495
5	0.97929	0.84277	0.82059	0.7736	1	0.45108	0.12454	0.12499
6	0.46061	0.38015	0.41488	0.34895	0.45108	1	0.23449	0.23817
7	0.12435	0.12235	0.10118	0.12449	0.12454	0.23449	1	0.964
8	0.12479	0.12273	0.10088	0.12495	0.12499	0.23817	0.964	1
9	0.68096	0.74968	0.54924	0.8077	0.69148	0.43203	0.15595	0.15745
10	0.67004	0.57712	0.5653	0.52975	0.6782	0.65871	0.20235	0.2053
11	0.8275	0.68295	0.74533	0.62689	0.81036	0.55664	0.13141	0.13206
12	0.78347	0.87388	0.63944	0.82499	0.79664	0.42298	0.14092	0.14192
13	0.67323	0.55563	0.60639	0.51003	0.65929	0.67022	0.1228	0.12316
14	0.97231	0.84882	0.83307	0.77915	0.98505	0.44786	0.12261	0.123
15	0.12479	0.12273	0.10088	0.12495	0.12499	0.23817	0.964	1
16	0.83664	0.69049	0.75356	0.63381	0.81931	0.550560	0.12926	0.12984
17	0.77322	0.66884	0.79762	0.60622	0.75788	0.52015	0.10853	0.10847
18	0.60001	0.727	0.53332	0.79301	0.6127	0.6127	0.10652	0.1105

50*50

Fig. 4: Similarity matrix

(Fig. 2), which is quite large and very sparse. Hence, it is essential to use the SVD method to perform dimension reduction. The reduced document matrices U, S and V follows (Fig. 3). In V, for n number of documents, this matrix contains n number of rows holding eigenvector values. Each of these rows then holds the coordinates of individual document vectors. Now, this matrix is taken for finding the similarity that exists between the documents. First, the commonly used cosine measure method is applied to construct the similarity matrix. Then, the fuzzy measure is used and obtained the similarity matrix (Fig. 4).

CONCLUSION

The goal of information retrieval is to make it easy for the user to obtain automatically the data relevant to the user on request. The system models documents and user request as vectors using the vector space model. The performance of the vector space model depends on the term weighting schemes, in other words, the functions that determine the components of the vectors. Some popular term weighting schemes together with few new term weighting schemes when applied provide positive results, the conclusion being that, "lgn" and "ls*mg*n"

term weighting schemes accomplished better than all the others. Regarding the similarity measures, fuzzy similarity measure is suggested for document retrieval which eases the burden and stress of time involved in document retrieval and provides more accurate similarity values facilitating an easier interpretation of the results than the cosine similarity measure.

REFERENCES

Alexander Strehl, 2002. Relationship-based Clustering and Cluster Ensembles for High-dimensional Data Mining. Dissertation for the Degree of Doctor of Philosophy.

Buckley, C., 1993. The Importance of Proper Weighting Methods. In ARPA Workshop on Human Language Technology, Princeton, NJ, Morgan-Kaufmann, pp: 349-352.

Egghe, L., 2004. Vector retrieval, fuzzy retrieval and the universal fuzzy IR surface for IR evaluation. Inform. Proc. Manage., 40: 603-618.

Erica Chisholm and Tamara G. Kolda, 1999. New term weighting formulas for the vector space method in information retrieval. Technical Report ORNL-TM-13756, Oak Ridge National Laboratory Oak Ridge, TN.

Harman, D., 1992. Ranking Algorithms in Information Retrieval: Data Structures and Algorithms, W.B. Frakes and R. Baeza-Yates (Eds.). Prentice Hall, Englewood Cliffs, NJ., pp: 131-151.

Jiawei Han and Micheline Kamber, 2003. Data Mining Concepts and Techniques, Moegan Kaufmann Publishers.

Porter, M.F., 1980. An algorithm for suffix stripping Program, 14: 130-137.

Rui Xu, Wunsch, D. II., 2005. Survey of clustering algorithms. Neural Networks IEEE. Trans., 16: 645-678.

Salton, G., C. Yang and A. Wong, 1975. A Vector-Space Model for Automatic Indexing. Commun. ACM., 18: 613-620.

Salton, G. and Buckley, 1988. Term weighting approaches in automatic text retrieval. Inform. Proc. Manage., 24: 513-523.

Salton, G., 1989. Automatic Text Processing: The Transformation, Analysis and Retrieval of Information by Computer, Addison-Wesley, Boston, MA.