Query Explication by Semantic Approach

Shruti Gupta[1]

[1]M.Tech Scholar, JaganNath University, Jaipur, Rajasthan, India

Abstract: ***In the present scenario a large volume of data is available related to each and every topic. This increase in the amount of data has led to a situation where users are flooded with information and have difficulty sifting through the reams of material, much of which is not relevant to them. This is commonly referred to as the problem of information overload. So the main focus is to add a new dimension to Internet- Searching and that is to apply semantic aspects towards it. In precise words, "the search must be what user wish, not what he/she types". This paper presents the Latent semantic analysis (LSA) implementation on the terrier which provides a semantic relation between the terms and the documents.***

Keywords: Semantic, LSA, information overload, Internet-Searching.

1. Introduction

A typical day of million web users all over the world starts with a simple query. The quest for information on a particular topic drives them to search for it, and in the pursuit of their info the terms they supply for queries varies from person to person depending on the knowledge they have[1]. One of the most popular and widely used algorithms for extracting documents which are similar to a query document is TF-IDF [8], [6]**.** It measures the similarity between documents by comparing their word-count vectors. The similarity metric weights each word by both its frequency in the query document (Term Frequency) and the logarithm of the reciprocal of its frequency in the whole set of documents (Inverse Document Frequency). But this approach was not successful as the retrieval according to meanings of the words was not possible. It computes document similarity directly in the word-count space, which can be slow for large vocabularies. It assumes that the counts of different words provide independent evidence of similarity. It makes no use of semantic similarities between words. To remedy these drawbacks, numerous models for capturing low dimensional, latent representations have been proposed and successfully applied in the domain of information retrieval. A simple and widely-used method is Latent Semantic Analysis (LSA**)**, which extracts low-dimensional semantic structure using SVD decomposition to get a low-rank approximation of the word-document co-occurrence matrix [5]. This allows document retrieval to be based on "semantic" content rather than just on individually weighted words.

.

2. Abstract Model of Information Retrieval

Information retrieval is the process of obtaining information from a collection of documents. The information which is retrieved can be relevant or non relevant [2] .The information retrieval system helps the users in finding the information they need. It does not explicitly return information or answer questions. Instead, it informs on the existence and location of documents that might contain the desired information [7].

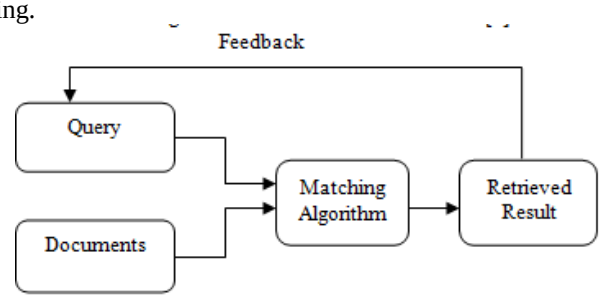

Figure 1 Abstract model of Information Retrieval

3. Semantic Approach

Efforts to incorporate semantic information into text processing systems date back nearly half a century. Over the years, designers have followed various approaches to integrating some degree of semantic processing into their information retrieval systems [3].

1 Auxiliary Structures
2 Local Co-Occurrence Statistics
3 Latent Semantic Indexing

3.1 Auxiliary Structures

Controlled vocabularies, or auxiliary structures, such as dictionaries and thesauri, allows narrower terms, broader terms, and related terms to be incorporated into queries [3]. Controlled vocabularies are one way to overthrow some of the most severe constraints of Boolean free-text keyword queries is multiple words that have similar meanings (synonymy), and words that have more than one meaning (polysemy). Synonymy and polysemy are often the cause of mismatches in the vocabulary used by the authors of documents and the users of text retrieval systems.

3.2 Co-Occurrence Statistics

Information retrieval systems using this method count the number of times pairs of terms appear together (co-occur) within a range of terms or sentences (for example, ±5 sentences or ±50 words) within a document. This approach is simple, but it only highlights a small portion of the semantic information in a collection of text.

3.3 Latent Semantic Analysis (LSA)

It is a mathematical/statistical method that can be used to decide similarity of meaning of terms and paragraphs by analyzing huge text [3].It does not use any artificial intelligence method or a natural processing method. It tries to explore the meaning of the words and about the topic. It has an added advantage of the semantic structure i.e. detection of the relevant documents on the basis of the queries**.** Each cell contains the frequency with which the word of its row appears in the passage denoted by its column [4].

Steps in LSA

1. The first step is to represent the text as a matrix in which each row stands for a unique word and each column stands for a text passage or other context. Each cell contains the frequency with which the word of its row appears in the passage denoted by its column.
2. Next, Singular value decomposition (SVD**)** [9] is applied to the matrix. In SVD, a large term by document matrix is disintegrated into a set of orthogonal matrices and a diagonal matrix. Queries are represented as pseudo-document vectors formed from weighted combinations of terms and documents. The SVD of matrix A is written as [8], [9].

$$A= U SV^T (1)$$

Where A is t×d term by document matrix is orthogonal matrix, S is a Diagonal Matrix, V is an orthogonal matrix and k is the rank. By changing all but the top *k* rows of S to zero rows, a low rank approximation to A called Ak is obtained

$$A_{k=} U_k S_k V_k^T (2)$$

Where *Uk* is the *t*×*k* term-by-concept matrix, S*k* is *k*×*k* concept-by-concept matrix, *Vk* is *k*×*d* concept-by-document matrix. The rank of *A* has been lowered from *r* to *k*. This low rank approximation removes redundancy from original data and allows us to uncover latent semantics is relations among terms as well as documents. Queries are formed into pseudo-documents that specify the location of the query in the reduced term-document space.

$$q=q^T U_k S_k^{-1} (3)$$

or

$q_c=q'*M$ where M is the product of U_k and S^{-1}_k

4. Corpora and Experiment

4.1 Dataset and Queries

The experiments are carried 3 documents
d1: Shipment of gold damaged in a fire.
d2: Delivery of silver arrived in a silver truck.
d3: Shipment of gold arrived in a truck.
q="gold silver truck"

4.2 Work Done

The steps involved in the experiment are as follows:-
1. Creating a Term document and query matrix for the document
2. Calculating the Singular Value Decomposition
3. Applying the dimensionality reduction on the matrix
4. Executing Query and Result of Query

Table 1: Term-document matrix and query matrix

Terms	d1	d2	d3	q
a	1	1	1	0
arrived	0	1	1	0
damaged	1	0	0	0
delivery	0	1	0	0
fire	1	0	0	0
gold	1	0	1	1
in	1	1	1	0
of	1	1	1	0
shipment	1	0	1	0
silver	0	2	0	1
truck	0	1	1	1

Even with a collection consisting of just 3 documents the A matrix is not that small, so we better resource to software to simplify calculations. The Blue bit Matrix Calculator is used for matrix construction.

The above term document matrix is fed as input in the Blue Bit Matrix Calculator.

The

U:
0.420 0.075 -0.046
0.299 -0.200 0.408
0.121 0.275 -0.454
0.158 -0.305 -0.201
0.121 0.275 -0.454
0.263 0.379 0.155
0.420 0.075 -0.046
0.420 0.075 -0.046
0.263 0.379 0.155
0.315 -0.609 -0.401
0.299 -0.200 0.408

S:
4.099 0.000 0.000
0.000 2.362 0.000
0.000 0.000 1.274

V^T
0.494 0.646 0.582
0.649 -0.719 0.247
-0.578 -0.256 0.775

Then the Dimensionality Reduction is applied on to it with K=2 that is reducing the 3 matrices to 2 column.

The Query Vector and the Document Vector are calculated by the following equations:-

$$d = d^T U_k S_k^{-1} (4)$$
$$q = q^T U_k S_k^{-1} (5)$$

After the calculation the document vector so obtained is

d1 (-0.494, 0.649)
d2 (-0.645, -0.719)
d3 (-0.581, 0.246)

and the query vector is -0.214 -0.182

Finding the similarities by Cosine Similarity:-

Sim= (q.d)/|q|.|d|
d1=-0.0541
d2= 0.9910
d3= 0.4478

So the Document Ranking is d2>d3>d1.

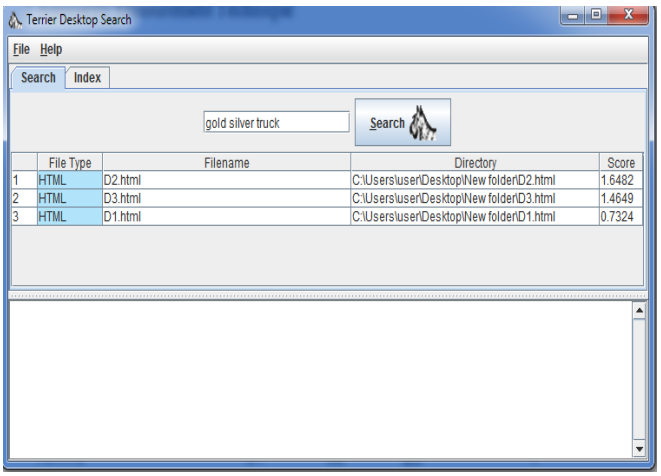

Figure 2: Screenshot of the Terrier

5. Conclusion

LSI has proven to be an optimal solution for a wide range of conceptual matching problems. One consequence of LSI processing is the establishment of associations between terms that occur in similar contexts. As a result, queries against a set of documents that have undergone LSI will return results that are conceptually similar in meaning to the query even if they don't share a specific word or words with the query.

References

[1] R.B. Yates, B R.Neto, Modern Information Retrieval Pearson Education, 1999.
[2] N.J. Belkin, W.B.Croft, Information Filtering and Information Retrieval: Two sides of the same coin?
[3] Communications of the ACM, 35, 1992, 29–38.
[4] Price, R., and Zukas, A., Application of Latent Semantic Indexing to Processing of Noisy Text, Intelligence and
[5] Security Informatics
[6] F. Cacheda, V. Plachouras, and I. Ounis. A case study of distributed information retrieval architectures to index one
[7] Terabyte of text. Information Processing & Management 41(5):1141{1161, 2010}
[8] A. Kontostathis, W.M. Pottenger, A Mathematical View Of Latent Semantic Indexing: Tracing Term Co Occurrences, Lehigh University Technical Report, LU-CSE-02-006, 2002
[9] G. Salton and C. Buckley. Term-weighting approaches in automatic text retrieval . Information Processing and
[10] Management, 24(5):513–523, 1988
[11] Salton, G. and M. McGill (1983). Introduction to Modern Information Retrieval. McGraw-Hill
[12] Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent Dirichlet Allocation. Journal of Machine Learning
[13] Research, *3*, *993*-1022 , Lecture Notes in Computer Science, Volume 349 5, Springer Publishing, 2005, pp . 602-603
[14] Ch. Aswani Kumar, Ankush Gupta, Mahmooda Batool and Shagun Trehan Latent Semantic Indexing-Based
[15] Intelligent Information Retrieval System for Digital Libraries . Journal of Computing and Information Technology - CIT 14, 2009, 3, 191–196

Author Profile

Shruti Gupta completed B.E (Hons) in 2009 in Information Technology and is an M.Tech Scholar .Currently Working as Assistant Professor in J.N.I.T, Jaipur in Computer Science Department. She has 3.5 years of experience.