Analysis of Medical Publications with Latent Semantic Analysis Method

José Román Herrera-Morales, Liliana Ibeth Barbosa-Santillán
Information System Department
University of Guadalajara, México
Email: {rherrera, ibarbosa}@cucea.udg.mx

***Abstract*—This article presents a review of the Latent Semantic Analysis (LSA) method used to extract knowledge from large sets of text documents, describing its origins, main applications, basic operation and dimensionality optimization. To evaluate its performance and usefulness in identifying semantic relatedness a series of experiments were conducted with various collections of texts, varying number of files that were part of each corpus and using different indexes. It was shown that LSA can serve as a mechanism for grouping and classifying documents that are related to the themes, in particular in obedience, to the search expressions according to their semantic relevance. It was also evident, however, that the computational performance of LSA will deteriorate as more files are added to generate indexes, since index and search response times increased significantly.**

Keywords*— *Latent Semantic Analysis; Semantic Relatedness; Semantic Relevance; Text Processing.

I. INTRODUCTION

This article is an analysis of Latent Semantic Analysis (LSA) [1], one of the most frequently used methods in search engines, text comparators, and recommender systems, since it allows the extraction of meaning and non-obvious relationships of terms in large sets of text documents. The idea is to describe the utility of LSA when applied to collections of texts from the medical field, to find documents that are the most relevant or similar in terms of content (semantic relatedness) according to the search terms. LSA represents an alternative to the need for human experts to analyse and digest information and is very important to apply it to the area of Health Sciences, one of the areas in which a large amount of scientific content is generated every day.

The structure of this document is as follows: the next section is a review of the concepts of LSA and the relationship between the application of matrix decomposition techniques of linear algebra such as the Singular Values Decomposition (SVD). Section III describes the experimentation phase, outlining the collection of medical documents, the implementation of the method in a LSA prototype to perform several tests and the integration of test scenarios to carry out the semantic relatedness test. Section IV describes the major results obtained, from the time of indexing and responding to queries, to the relevance in similarities of the meaning in documents. Finally, conclusions and comments about the results obtained are included in Section V.

II. LATENT SEMANTIC ANALYSIS

LSA is a computational model of human knowledge representation that approximates the ability to make judgments of semantic relationship, which is based on a very simple premise, namely, that the similarity in the meaning of two words can be induced by how they are used in texts [1]. By means of this principle, words and text are created in a specific domains. LSA examines the frequency in a set of texts and then uses semantic relatedness in order to build the matrix decomposition. In a nutshell, LSA is a knowledge representation model, which is based on the patterns of word usage in a range of documents. This set of documents is commonly called a corpus and the mapping between documents and terms is called Latent Semantic Space [2].

The following subsections address issues related to LSA that include: its origin, the basic operation of LSA and its relationship with SVD, the importance of optimizing the dimensionality of the matrices, and finally, the main areas of application of LSA.

A. Origin and first applications

LSA was released under patent #4,839,853 of the U.S. Patent Office issued on June 13, 1989 to Bell Labs researchers Deerwester, Dumais, Furnas, Harshman, Landauer, Lochbaum and Streeter, and was originally used as a mechanism to support tasks of Information Retrieval [3] [4]. It was mentioned as Latent Semantic Indexing (LSI) in order to use techniques of dimension reduction for improving the indexing process of textual content [5]. Subsequently, Landauer and Dumais, who were interested in human learning and how people learn new vocabulary from the texts that they read [6], proposed the LSA as a new theory regarding acquisition, induction and knowledge representation to reflect the similarity of words and passages of text, making use of the analysis of a large corpus of natural text. They observed that, by inducing global knowledge indirectly from a co-occurrence data locally on a large body of characteristic texts, the LSA can acquire knowledge of the entire English vocabulary in a manner comparable to the way a child learns. After reading texts, children can learn new words every day and after several readings can apparently understand the meaning of many other words that they did not know before. This feature of human language learning has been a topic of debate and research interest. In ancient times, it was known as Platon's problem, i.e., how people can know much more than they have been exposed to. Platon suggested that people had all this knowledge within themselves and only needed small patterns or guides to be able to produce it. LSA means analogous hidden meanings can be extracted when information processing of a collection of texts is performed.

 ISBN: 978-1-61208-311-7

B. Basic operation

The LSA method consists of a series of basic steps for extracting meaning from a collection of documents. First, it generates a term-document matrix, where each row represents the words in the whole collection of texts and the columns represent the documents. In this first step those words whose occurrence in the collection of documents is too frequent or too infrequent should be eliminated, as should words that do not add any value, so-called stop-words. Second, an algorithm to calculate the weights for each of the cell-matrix document terms is applied, this for emphasis of the words according to a certain domain. Finally, the SVD process is applied. As a result of this process, three partial orthogonal matrices are produced: the term-matrix (commonly called matrix U or Left Singular Values), the document-matrix (commonly called matrix V, or Right Singular Values) and the diagonal matrix S, whose main diagonal contains singular values and other positions zeros [7]. Fig. 1 shows the original matrix A (term-document matrix) and the resulting SVD matrices.

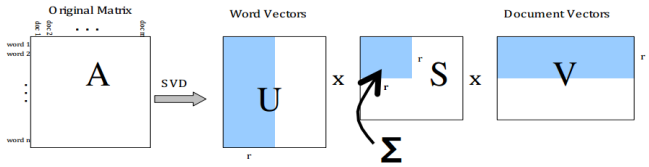

Fig. 1: SVD process applied to matrix A that produces three orthogonal partial matrices as a result (source: Kireyev & Landauer, 2011) [8].

C. Optimal dimension reduction

Reducing the number of dimensions by applying minimal SVD decomposition significantly reduces the noise and the amount of data, memory and processing time required to obtain results with LSA. This process is called optimization dimensionality [9] and involves finding the K-th dimension (the columns that represent collections of documents) for the best K-dimensional approximation of the original matrix. Thus, the document collection is represented by a K-dimensional vector space derived by SVD. In many cases, the value of K is much smaller than the number of terms that are present in the matrix of term-document, but for application related simulation language learning, it was found that the optimum value of K is in a range of 300 +/- 50 [6], [9], [10] and validated with a formal study applied similarity in meaning tests for text samples from the Groliers Academic American Encyclopaedia described by Landauer and Dumais [6]. Fig. 2 shows the original graph of this study where it can be seen that there are sufficient values close to 300 in the number of dimensions to be considered, since it is this range which gives the best similarity in meaning.

D. Areas of application

LSA can greatly improve the extraction and representation of knowledge in the domain of human learning to represent objects and contexts present but can also be applied in situations with a large volume of data, such as Data Mining. Wolfe and Goldman [1] found LSA useful in a processing and text analysis, such as quality assessment and summary

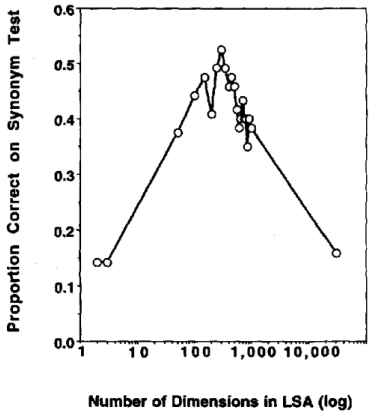

Fig. 2: The effect of K-dimensions retained in LSA-SVD simulations of meaning similarities. K-dimensions is in log scale (taken from Landauer & Dumais, 1997).

trials, finding differences or similarities between texts verifying internal coherence, and in identifying the original source of students' work, among many others. All these applications have had very good results, and the reliability of LSA has been so good that it is comparable to human experts.

The following section describes in detail the experiments that have been conducted to evaluate the operation, performance and utility of the LSA method in identifying semantic concordance using like a corpus a collection of texts containing abstracts of articles in health sciences field. The medical field is one of the fields of research that is growing more rapidly and all new medical information is being published everyday [11]. Hence, the importance in working to generate mechanisms that allow this scientific community to have better access to this large amount of resources.

III. EXPERIMENTS

This stage of experimentation, where the semantic relatedness tests were carried out, was divided into three main phases: (1) Getting the text collection, in order to obtain a raw material that represents items in the health sciences field, (2) implementation of the LSA method, a tool coded in C# language, and (3) the definition of test scenarios that included a series of searches in several corpora with different characteristics in terms of the number of files and the value of K, to optimize the LSA process.

A. Text collection

The first step before testing LSA was to generate multiple text files to serve as the corpus or data source. This information can be obtained with the OAI-PMH Service from PubMed Central (PMC-OAI) [12] that provides access to the metadata for all items in its collection. The Open Archives Initiative Protocol Of Metadata Harvesting (OAI-PMH) [13] is a standard protocol for the collection of metadata records designed to be shared openly and freely, and it is promoted by the Open Archive Initiative and it is based on the exchange of XML messages on a transport service such as HTTP.

Once an excellent source of information in the medical field has been identified and there is a reliable way to obtain

 ISBN: 978-1-61208-311-7

it, a .Net TCP client application is used to make requests to the OAI-PMH server in PubMed Central in order to download the metadata records. Given the characteristics of PMC-OAI service, the resulting records were delivered in Dublin Core simplified format and provided more than 300,000 metadata records through the OAI-PMH harvester client. For each record retrieved, the OAI-PMH harvester client generated a text file in a local directory, and each file contained the following information fields: title, authors, abstract, date of publication, journal, publisher and the ID assigned by the PMC-OAI service.

These files were metadata items that were filtered and identified as research articles and were discharged in chronological order by publishing date, the most recent first, i.e., April 2013 up to July 2008. Although the PMC portal states that it had 2.7 million articles, it stopped downloading them because for experimentation conducted in this article was considered as the limit 1000 files due to the considerable time that indexing is required for this amount. This behavior is described in more detail in the results section.

B. Implementation of LSA method

Various options for implementing SVD were reviewed, such as Bluebit - Online Matrix Calculator that allows online calculations of small matrices and other tools much more complete as the ”R”; which contains a specialized package for LSA. But, familiarity with .NET platform and the ability to adapt and customize the code, as well as to select a local folder with n number of files as a data source, were the main reasons for the implementation in C# by Anup Shinde [14] should be selected. This used the open-source libraries DotNetMatrix [15] for all tasks concerning the matrix algebra including SVD decomposition.

The main features of the prototype in C# are: set configuration options, maintenance of indexes and use of queries to verify the consistency of a search expression in the corpus. In the settings section, the user can define a local folder where it takes the collection of documents to the index, and set the value of K to be used for dimensionality reduction of the resulting matrices of SVD. For queries, the results are provided in two ways: first, as an individual list of documents ranked according to their percentage of semantic relatedness with the search expression, and the second, as a view grouped into ranges of percentage of relevance to know the quantity of documents that fall into each category. Additionally, options were enabled to store the full and reduced SVD matrices, as well as functionality of exporting to CSV format. In Fig. 3, a screenshot of the GUI of this prototype is shown.

The workflow of this implementation can be analysed by dividing it into two main groups: first, the LSI index generation for test collection of documents, and second, the search process in the document collection. This last part includes how to present the results in the GUI, so that they can be interpreted in a simpler way.

C. Definition of test scenarios

Before the experiment started, several indexes were generated with different numbers of files so that semantic matching tests could be performed under different test scenarios.

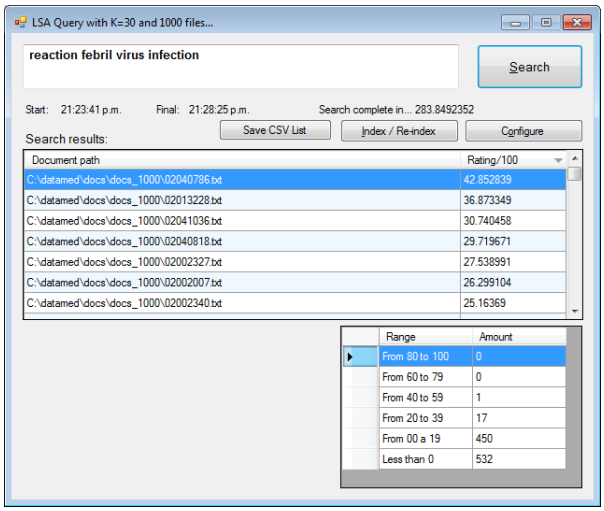

Fig. 3: The GUI of implementation of LSA method in C#

Different indexing times were counted, considering first small numbers of files, starting with 10, 100, 150, 200,250, 300, 400, 500, 600, 700, 800, 900 and 1000 files. For each of these quantities two indexes were generated, one considering the value of K as 10% of the files and one with K= 50%. Files that were considered in each set were selected indexed over 300,000 files retrieved with OAI-PMH harvester and ordered in ascending order according to their name in the local folder. The largest index always includes in its entirety all of the previous index files.

Several special cases were presented, when K= 50% represented more than 300 files (amount that exceeds the recommended optimal value). These cases generated new indexes with K constant values such as 250 and 300, when applied to the corpus translated into 700, 800, 900 and 1000 files. In this way the maximum dimensionality considered was 250 and 300.

To provide consistency in terms of the evaluation and comparison of the results that were obtained, a list of queries was generated and applied in the same way to each of the test events with the several LSI indexes. The list of queries, formed by sequences of non-sorted terms, is as follows:

Query 1: reaction febril
Query 2: reaction febril virus infection
Query 3: tissue epidermis skin carcinogen
Query 4: cancer tumor carcinoma
Query 5: cancer tumor carcinogenesis

IV. Results

The results are described in terms of three main groups, and are referring to: (a) the indexing times, (b) the average response times for queries, and (c) the semantic relatedness tests.

A. Indexing time

Fig. 4 shows that there is no significant difference in time indexing for indexes with fewer than 600 files, but more than 600 means an increase in time indexing when considering a K

 ISBN: 978-1-61208-311-7

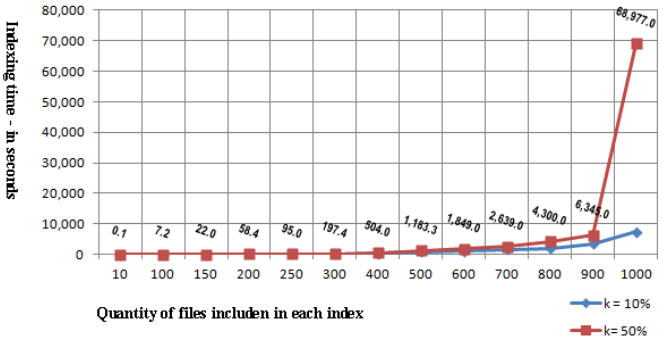

Fig. 4: Statistics for LSI time with K-values in different indexes

with a value of 50% of the number of files. A very important fact is that for 1000 files, considering K= 50% (500 columns for reduced SVD), indexing time increased very considerably; in fact, when there were 900 files, 6345 seconds (105 minutes) increased to 68977 seconds (1150 minutes; or 19 hours or almost a full day indexing); this means that from one index to another it grew in more than 10 times the necessary time to be able to index all the documents. By contrast, with K= 10% indexed time observed normal growth.

Additional indexes with 600, 700, 800, 900 and 1000 files were performed, considering the value of K as constant values, 250 and 300, values considered optimum [6], [9], [10]; this was done to reduce the dimensionality of the resulting matrices on SVD.

Fig. 5 includes indexing times; when using K= 300, as seen, for indexes many files with a greater double the recommended value of K and has a slight increase in the case of 1000 files also begin to increase but not as disproportionately as in the case where K= 50%. In the latter index, for 1000 files it took 7392 seconds for K= 10% (K= 100), 22605 seconds for K= 300 and the aforementioned 68977 seconds for K= 50% (K= 500).

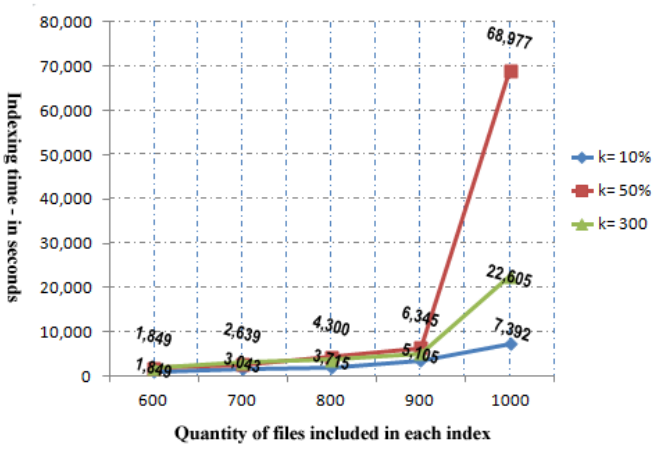

Fig. 5: Comparison of reduced indexing time with K= 300

Considering the times obtained in the previous 900 indexed files, the rate of increase over the previous indexes increased approximately 2x for K= 100, 4x for K= 300 and 10x for K= 500. When it became clear that, when the number of files to be indexed is close to or greater than 1000, the indexing time increases significantly, the decision was taken to set this value as the file limit for these LSA semantic relevance tests, so that all queries that are described in the next section treat 1000 as the maximum number of files for the larger index.

B. Average response times for queries

After the indexing process, the verification queries came where each repeated one defined and executed only 12 different indexes (Fig. 6), in which the number of files and the respective value of K varied.

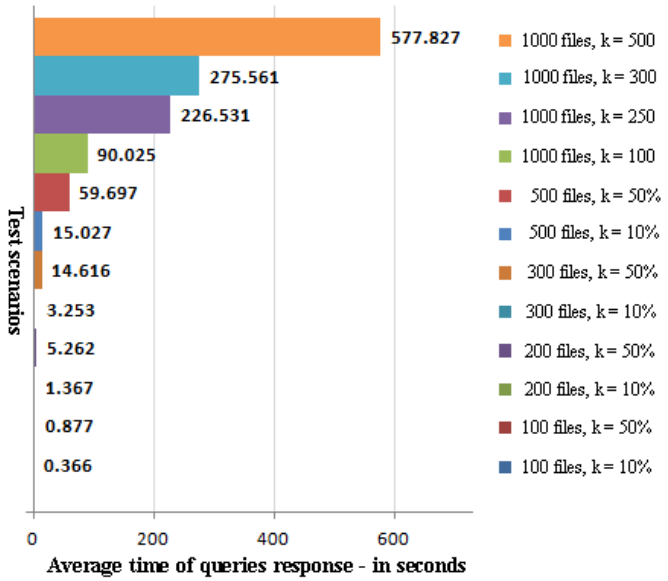

Fig. 6: Average response time for queries with different indexes varied.

Fig. 6 shows a graph with the respective average response times and the remarkable time it takes to answer a query in the case where K= 500 to 1000 files; it takes 577 seconds (nearly 10 minutes) to deliver the results on screen.

C. Semantic relevance tests

For each of the queries, response times were recorded and the GUI of the prototype in C# displayed the most relevant files according to the semantic coherence of its content. Furthermore, a clustering result was generated when files were included in six ranges relevant percentage according to the query made. The ranges were 80 to 100%, 60 to 79%, 40 to 59%, 20 to 39%, 0 to 19% and less than 0%. To find out how many files corresponded to each of the ranges of relevance, the percentage obtained was recorded at each event; this in order to have a quantitative way to measure the semantic relatedness of each query.

For a better analysis of the results the data are presented in tables. In these tables, the columns represent each of the events in which searching indexes have been used with different values of K; in the first columns, the values of K are expressed in % of files of the corpus, while in the last columns there are a certain number of files (100, 500, 250 and 300 files). On the other hand, the first rows represent the amount of files

 ISBN: 978-1-61208-311-7

that falls in each range of percentages of relevance, and the last two rows show the percentages of two of the files more relevant for each query. When an "*" appears in the cell it means that the examined file does not appear in the Top Ten. When the cell value is displayed in bold and shaded it means that it occupied the first place.

Table I shows then concentration of these results in relation to search expression "febrile reaction". It show that very few files, accumulated events, fell within the range of 40 to 59% of significance (92 files), while 26 files were in the range 60 to 79%, and only five in the range of 80 to 100%. View the last semi-right column. These figures indicate that it has a small number of files whose content is related to febrile reactions. One file in particular identified as "02002007", in more than half of the search events, reached first position in the ranking of relevant files. This is shown in the last row of Table I.

TABLE I: CONCENTRATED DATA RESULTING FROM QUERY 1

Query1: "reaction febril"

	K-values for Reduced SVD												
Range	10%	50%	10%	50%	10%	50%	10%	50%	100	500	250	300	
80 - 100	4	0	1	0	0	0	0	0	0	0	0	0	5
60 - 79	9	1	7	1	3	1	2	0	2	0	0	0	26
40 - 59	24	1	19	3	12	3	10	3	8	1	4	4	92
20 - 39	12	6	24	1	41	2	52	7	29	8	13	13	
0 - 19	42	92	126	195	106	131	196	226	467	479	454	448	
less than 0	9	0	23	0	138	163	240	264	494	512	529	535	
# files in corpus	100	100	200	200	300	300	500	500	1000	1000	1000	1000	
02002007 - - %	*	*	73	**68**	53	**66**	58	**55**	**63**	**41**	**51**	**47**	
01999693 - - %	**88**	**78**	67	46	*	46	*	34	*	28	40	38	

Query2 "reaction febril virus infection" was very similar to query1, two more words being added for a more precise search in medical articles for febrile reactions, but in this case caused by viral infection. Table II shows the concentrated results.

The results of Table II evidence how a very reduced group of files was in the first three ranks (last semi-column to the right) which shows the specialization of their contents in accordance with the search expression. For this query2, the file "02002007" did not achieve the top position in any of the tests, and instead the file "01997182" reaches the first places only in the first events. A different file was the most similar in terms of content, it was the file "02040786" which in the last four events (last columns that represent corpus with 1000 files) was located in the first position of relevance with 66%, 33%, 45% and 43%, respectively.

The effect of specialization has been evidenced in the query3 "tissue epidermis skin carcinogen" when more precise terms were added to the search expression. Table III shows that the file "02001792" always was ranked in the first place, also in the last columns with the corpus of 1000 files, shown stability in results, because the average relevance was 61% +/- 3% (with K=100, 250 and 300); and except in the case where K=1000 the relevance fell to 47%. Here is evidence that a greater amount of items in the index does not help to improve its effectiveness, but on the contrary this distorts the result.

Table V shows data very similar to the previous query results (Table IV). The difference between query4 and query5 was only the third word ("carcinoma" and "carcinogenesis")

TABLE II: CONCENTRATED DATA RESULTING FROM QUERY 2

Query2: *"reaction febril virus infection"*

	K-values for Reduced SVD												
Range	10%	50%	10%	50%	10%	50%	10%	50%	100	500	250	300	
80 - 100	5	0	1	0	1	0	0	0	0	0	0	0	7
60 - 79	6	3	11	0	5	0	2	0	2	0	0	0	29
40 - 59	8	2	9	4	12	2	16	0	15	0	1	1	70
20 - 39	8	2	19	11	25	14	25	16	42	13	22	17	
0 - 19	66	93	155	185	104	114	192	209	416	479	456	450	
less than 0	7	0	5	0	153	170	265	275	525	508	521	532	
# files in corpus	100	100	200	200	300	300	500	500	1000	1000	1000	1000	
02040786 - - %	*	*	*	*	*	*	*	*	**66**	**33**	**45**	**43**	
01997182 - - %	92	**73**	70	**51**	*	**45**	57	34	*	22	*	*	

TABLE III: CONCENTRATED DATA RESULTING FROM QUERY 3

Query3: "tissue epidermis skin carcinogen"

	K-values for Reduced SVD												
Range	10%	50%	10%	50%	10%	50%	10%	50%	100	500	250	300	
80 - 100	3	1	2	1	1	0	0	0	0	0	0	0	8
60 - 79	4	1	1	1	2	2	6	1	1	0	1	0	20
40 - 59	4	0	9	0	18	1	3	1	6	1	1	2	46
20 - 39	17	1	26	1	42	2	43	7	39	6	14	12	
0 - 19	62	97	143	95	100	133	206	232	479	480	451	464	
less than 0	10	0	19	102	137	162	242	259	475	513	533	522	
# files in corpus	100	100	200	200	300	300	500	500	1000	1000	1000	1000	
02001792 - - %	**96**	**91**	**93**	**87**	**80**	**75**	**77**	**65**	**64**	**47**	**63**	**58**	
01997142 - - %	95	79	90	75	69	68	65	58	51	34	56	50	

which a human expert would interpret them as equivalent. In this case, the results show that both queries yield nearly identical results to the four first events according to the corpus considering more files, (events that are further to the right, the results are more specialized and even grow in a few percentage points of semantic relatedness in case the file "01997145" which rises from 41% to 50% with K=250, and 39% to 46% with K=300, in both cases with the index with 1000 files.)

TABLE IV: CONCENTRATED DATA RESULTING FROM QUERY 4

Query4: "cancer tumour carcinoma"

	K-values for Reduced SVD												
Range	10%	50%	10%	50%	10%	50%	10%	50%	100	500	250	300	
80 - 100	5	1	5	0	8	0	2	0	1	0	0	0	22
60 - 79	2	2	2	2	2	0	15	0	6	0	0	0	31
40 - 59	4	1	7	3	8	5	8	2	21	1	4	2	66
20 - 39	10	3	11	7	13	4	10	13	27	15	30	27	
0 - 19	75	93	170	71	80	123	153	216	351	489	420	441	
less than 0	4	0	5	117	189	168	312	269	594	495	546	530	
# files in corpus	100	100	200	200	300	300	500	500	1000	1000	1000	1000	
01997145 - - %	96	**85**	94	**69**	**94**	**59**	84	**44**	69	31	41	39	
02002459 - - %	*	*	*	*	*	*	**92**	35	**83**	*	**49**	**41**	

Another interesting situation that can be noted from Table V is that file "02002459" practically does not appear in the first place of relevance; this is because the third word used in query5, "carcinogenesis", was a term even more technical in health sciences domain and therefore in its place was the file "2013034" that in the last two events was second in importance

 ISBN: 978-1-61208-311-7

TABLE V: CONCENTRATED DATA RESULTING FROM QUERY 5

Query5: "cancer tumour carcinogenesis"

	K-values for Reduced SVD												
Range	10%	50%	10%	50%	10%	50%	10%	50%	100	500	250	300	
80 - 100	5	1	5	0	8	0	2	0	1	0	0	0	22
60 - 79	2	2	2	2	2	1	15	0	10	0	0	0	36
40 - 59	4	1	4	1	6	3	8	4	17	1	7	4	60
20 - 39	10	3	12	7	15	7	6	8	20	12	20	21	
0 - 19	75	93	169	67	94	111	166	230	389	490	456	437	
less than 0	4	0	8	123	175	178	303	258	563	497	517	538	
# files in corpus	100	100	200	200	300	300	500	500	1000	1000	1000	1000	
01997145 - - %	96	83	96	75	92	70	87	54	78	38	50	46	
02013034 - - %	*	*	*	*	*	*	79	43	*	*	49	45	

and only by a few tenths of a percentage point missed first place.

A relevant fact that is presented in all scenarios and event searches is the demonstration of the optimal value of K, taking into consideration the corpus of 1000 files. Comparing the results of semantic relevance in different queries, we found that the similarity values of the file contents in first place were always more consistent with values of K= 250 or K= 300, while the value of K= 100 where most ranged up to differences of more than 10 percentage points with respect to the others. In the case of K=500 relevant results are generally within the average range, but we must not forget that for this value of K the indexing time is up to six times greater and the response time of other three times slower.

V. CONCLUSIONS

In the light of the results, several points should be made in relation to the operation and computational performance of the LSA method. It has been very interesting to have proven that the proper value of K is 250 and 300 as optimal solution.

Although LSA obtains semantic relatedness based on statistical techniques of frequency of terms, it has been possible to demonstrate that when search expressions include more terms, they are identified with greater precision and a more precise classification of documents dealing with the same topic, whereas documents that are not related are clearly separated from the rest of the group (Table I and Table II). Also, it was possible to verify that if it had a sufficient number of files that belong to the application domain, in this case the health science area, LSA can establish semantic relatedness to identify those words or terms that are equivalent for the same context (Table IV and Table V).

Finally, one of the major issues of the LSA method has been described as related to computational performance, the times of the indexing process and the corresponding time of execution of queries. While most files were added to the corpus, the indexing time was increasing considerably. Particularly, when the indexing of 1000 files with K = 500 took several hours to complete, and the search response times went from seconds to minutes (Fig. 5); in these cases the use of LSA is no longer convenient. Fortunately, the computer technology continues to evolve and it is probably that with greater computing power and applied techniques of clustering it will be possible to solve this type of problems.

ACKNOWLEDGMENT

The authors would like to thank to SEP-PROMEP México and University of Guadalajara for the financial support to make this research.

REFERENCES

[1] M. B. Wolfe and S. R. Goldman, "Use of Latent Semantic Analysis for predicting psychological phenomena: Two issues and proposed solutions", Behavior Research Methods, Instruments, & Computers, vol. 35(1), 2003, pp. 22–31.

[2] K. Christidis, G. Mentzas, and D. Apostolou, "Using latent topics to enhance search and recommendation in enterprise social software", Expert Systems with Applications, vol. 39(10), 2012, pp. 9297–9307.

[3] S. C. Deerwester, S. T. Dumais, G. W. Furnas, R. A. Harshman, T. K. Landauer, K. E. Lochbaum, and L. A. Streeter, "Computer information retrieval using Latent Semantic structure", June 13 1989. US Patent 4,839,853.

[4] Y. Tonta and H. R. Darvish, "Diffusion of Latent Semantic Analysis as a research tool: A social network analysis approach", Journal of Informetrics, vol. 4(2), 2010, pp. 166–174.

[5] S. T. Dumais, "LSA and information retrieval: Getting back to basics", Handbook of Latent Semantic Analysis, 2007, pp. 293–321.

[6] T. K. Landauer and S. T. Dumais, "A solution to plato's problem: The Latent Semantic Analysis theory of acquisition, induction and representation of knowledge", Psychological review, vol. 104(2), 1997, pp. 211–240.

[7] M. Ahat, S. B. Amor, M. Bui, S. Jhean-Larose, and G. Denhire, "Document classification with LSA and pretopology", Stud. Inform. Univ., vol. 8(1), 2010, pp. 125–144.

[8] K. Kireyev and T. K. Landauer, "Word maturity: Computational modeling of word knowledge", Proc. of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies. Association for Computational Linguistics, vol. 1, 2011, pp. 299–308.

[9] T. K. Landauer and S. T. Dumais, "Latent Semantic Analysis", Scholarpedia, vol. 3(11), 2008, pp. 4356.

[10] S. T. Dumais, "Improving the retrieval of information from external sources", Behavior Research Methods, Instruments, & Computers, vol. 23(2), 1991, pp. 229–236.

[11] M. Gillam, C. Feied, J. Handler, E. Moody, B. Shneiderman, C. Plaisant, M. Smith, and J. Dickason, "The Healthcare Singularity and the Age of Semantic Medicine". In The Fourth Paradigm Data-Intensive Scientific Discovery, Microsoft Research, 2009, pp. 57–64.

[12] PubMED Central. The Pubmed Central OAI-PMH Service (PMC-OAI). [Online] Available: http://www.ncbi.nlm.nih.gov/pmc/tools/oai/ (Last visit: 2013, July 12).

[13] OAI-PMH. Open Archives Initiative Protocol for Metadata Harvesting. [Online] Available: http://www.openarchives.org/pmh/ (Last visit: 2013, July 12)

[14] A. Shinde. Anup Shinde's web page, with LSI sample coded in C#. Available: http://www.anupshinde.com/latent-semantic-indexing/ (Last visit: 2013, July 12)

[15] P. Selormey. DotNetMatrix libraries. A basic linear algebra package for .Net, Available: http://www.codeproject.com/Articles/5835/DotNetMatrix-Simple-Matrix-Library-for-NET/ (Last visit: 2013, July 12)

 ISBN: 978-1-61208-311-7