Latent Semantic Analysis of Textual Data

Preslav Nakov

Latent Semantic Analysis of Text Information *The paper presents an overview of the usage of LSA for analysis of textual data. The mathematical apparatus is explained in brief and special attention if pointed on the key parameters that influence the quality of the results obtained. The potential of LSA is demonstrated on selected corpus of religious and sacred texts. The results of an experimental application of LSA for educational purposes are also present.*

Latent Semantic Analysis

The *Latent Semantic Analysis (LSA)* is a powerful statistical technique for indexing, retrieval and analysis of textual information used in different fields of the human cognition during the last decade. The method is fully automatic and does not use any preliminary constructed dictionaries, semantic networks, knowledge bases, conceptual hierarchies, grammatical, morphological nor syntactic analysers, etc. The general idea is that there exists a set of latent dependencies between the words and their contexts (phrases, paragraphs and texts). Their identification and proper treatment permits LSA to deal successfully with the synonymy and partially with the synonymy.

LSA is a two-stage process (see [2],[5][6]) and includes education and analysis of the indexed data. During the education phase LSA performs an automatic document indexing. The process starts with the construction of a matrix X whose columns are associated with documents, and the rows with terms (words or key-phrases). The cell (i,j) contains the occurrence frequency of term i in document j. The matrix X is then submitted to *singular value decomposition* (*SVD*) which gives as a result three matrices D, T (orthonormal) and S (diagonal), such that $X=DST^t$. Most of the rows and columns of D, S and T are removed in a way that the matrix $X'=D'S'T'$ is the least squares best-fit approximation of X. This results in the compression of the source space in much smaller one where we have only a limited number of significant factors (generally between 50 and 400). Thus, each term or document is associated a vector of reduced dimensionality, e.g. 100. It is possible to perform a sophisticated SVD, which speeds up the process by directly finding the truncated matrices D', S' and T' (see [1]).

The second phase is the analysis phase. Most often this includes the study of the proximity between a couple of documents, a couple of words or between a word and a document. A simple mathematical transformation permits to obtain the vector for a non-indexed text. This permits the design of a LSA based search engine processing natural language queries. The proximity degree between two documents can be calculated as the dot product between their normalized LSA vectors. The usage of other measures is also possible, e.g.: Euclidean and Manhattan distances, Minkowski measures, Pearson's coefficient etc.

Religious texts

This is a collection of English language religious texts we found at: *http://davidwiley.com/religion.html*. The whole documents collection includes 1424 files (21.7 MB) highly not proportionally distributed by count and volume among the different religions. The Old Testament for example includes 928 files (8.91 MB), the New Testament — 262 files (4.36 MB), and the Dead Sea scripts just 8 files (22 KB). As this disproportion can lead to significant space distortion we made a representative selection of 196 different documents (after removal of the HTML elements: 20443 different terms, 11140 of them used in at least 2 distinct documents), distributed in 11 categories: 4 kinds of apocripha (acts, apocalypses, gospels, writings), Buddhism, Confucianism, Dead Sea scripts, The Egyptian Book of Dead, Sun Tzu: The Art of War, Zoroastrianism, The Bible (2 subcategories: Old and New Testaments), The Quran and The Book of Mormons. The experiments were made in a 30 dimensional space in 4

different ways by applying or not logarithm and/or entropy. The results are shown on figure 1 in 5 different colors for the five correlation intervals: 87,5-100%, black color; 75-87,5%, dark gray; 62,5-75%, gray; 50-62,5%, light gray; 0-50%, white.

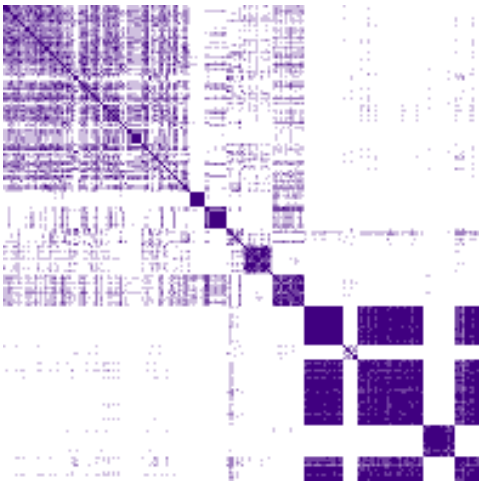

Figure 1. Correlation matrix

The dark rectangles in the main diagonal show the high correlation between texts belonging to the same religion. For example: the black rectangle from the bottom right corner contains texts from the Book of Mormons. To the left and up on the main diagonal can be found the Quran, then the Old Testament (The Bible), then come the Zoroastrian texts, The New Testament (The Bible), the Sun Tzu's Art of War, the Egyptian Book of the Dead and so forth. And the smooth rectangle in the upper left corner shows the relatively high similarity between all kinds of apocrypha present.

Let us consider the left matrix from figure 1 showing the original correlation matrix without any transformation applied. We can see several black rectangles outside the main diagonal. Because the matrix is symmetric we can observe only the part above it. The black rectangle in the middle of the lower right corner of the matrix shows the high correlation level between the New and Old Testament. This is something we expected because those are texts belonging to the same religion with common content style, describing common persons and events using the same or almost the same words. Because, as have been mentioned above, LSA deals very well with the synonymy it succeeds to identify the high proximity between the two parts of the Bible. In fact, as Figure 1 shows, it is impossible to distinguish between them.

What is interesting on that figure are the two other black rectangles in the last matrix rows showing the high proximity level between the two parts of the Bible and the Book of Mormons. One can consider this surprising. In fact a single look at the Book of Mormons is sufficient to see how close is its content to the Bible. Let now pass to the left matrix on Figure 2. We see that by applying the entropy transformation to the base matrix preliminary to SVD we can distinguish between the Old and the New Testament (The correlation is about 70% and the corresponding horizontal rectangle is no longer black but grey. Grey is the rectangle between the Book of Mormons and the Old Testament.) but we cannot distinguish between the New Testament and the Book of Mormons!

Results quality

Although it is comparatively old and well-studied technique the application of LSA is kind of art. There are two key factors that influence the quality of the results obtained: the proper choice of dimensionality and the application of suitable transformations on the raw matrix *X*. Unfortunately, there are no strict guiding rules how to tune these parameters and a research is needed for each particular case. In general, the application of logarithm for each of the elements of the matrix *X* leads to improved performance. Better results are expected if each row is divided by its corresponding entropy. (see [2]) So, the most frequent words (the noise) are weighted lower than the rare ones. Theoretically the best results are achieved when both transformations are applied one after the other, starting with the logarithm.

Figure 2 shows the correlation matrix when different preliminary transformations of the matrix *X* are applied before SVD. The results are very interesting. It is clear that the application of logarithm leads to noise reduction in the upper left matrix corner. The division of the rows by their entropy, as have been mentioned above, gives even better results and permits to distinguish between the parts of the Bible. What is interesting is

that the application of both transformations introduces a significant amount of noise and leads to very poor performance even regarding the original matrix. This result is consistent with our previous claim that there are no strict guiding rules when applying LSA.

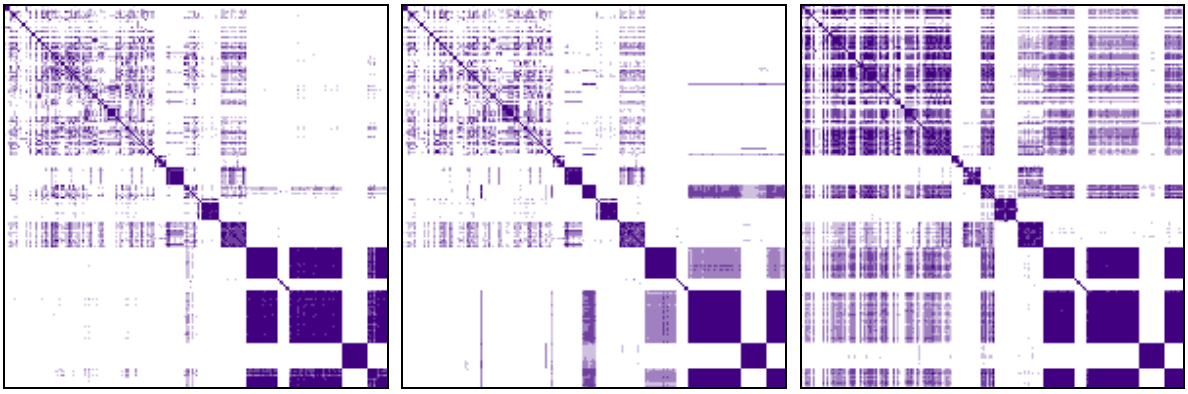

Figure 2. Correlation matrix: logarithm, entropy, entropy&logarithm

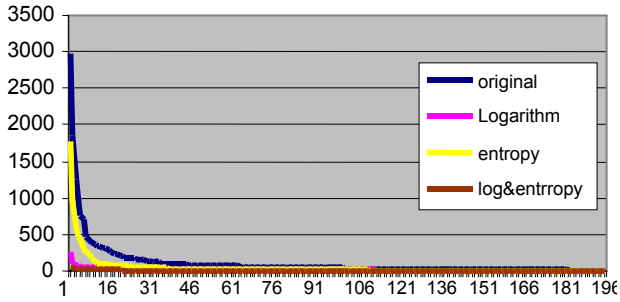

Figure 3. Choice of dimensionality

The second key parameter, as have been mentioned above, is the proper choice of dimensionality. Unfortunately this is an unsolved problem. Figure 3 shows all the 196 singular values (elements from the main diagonal of *S*) of the matrix *X*, ordered in decreasing order. We see that the curve flattens after 30. If we cut more values we lose important information, if we cut less we model the noise. The quality of the results is influenced by several other factors. It is a good idea for example to drop out the propositions, unions, adverbs, particles, auxiliary verbs, etc. (e.g. *the*, *of*, *a*, *an*, *and*), that do not bear important information. We used a special list of 938 stop words when preprocessing the religious texts. We dropped out also all the words having one or two letters, and those contained in just one document since they could not contribute to the discovery of a latent dependency between a couple of texts nor words. In case of a bigger corpus it may be convenient to apply stronger limitation in order to limit the distinct words count in the rows of *X.* In one of our experiments on a corpus of 1886 files (92.8 MB) we found 61 451 distinct words (a technical documentation *RFC* with a lot of specific terms and abbreviations). This number was unacceptable for us because our hardware was unable to perform a singular value decomposition of a matrix sized 1886 x 61541. (We were able to deal with it later under LINUX.) So, we were forced to remove all one- and two-letter words and to keep only those that are common to at least 5 different documents. As a result the word count climbed to 13 858 which allowed us to create the index successfully.

The quality of the results can be improved further in case we apply an automatic mechanism to correct some of the common mistakes similar to the well-known tool AutoCorrect provided by Microsoft Office. It is also very important to recognize the different grammatical or orthographic word forms and abbreviations as belonging to the same class (e.g. *cat* and *cats*, *center* and *centre*, *normalise* and *normalize*). A particular problem is caused by the different spellings of the same word. This may be a result of the usage of abbreviations or dashed/undashed versions of the same words: e.g. “preprocessing” vs. “pre-processing”. Similar problem is caused by a word composed of two or more other ones: e.g. “key word” vs. “keyword” or “key-word”. Some authors suggest that morphologically related words belonging to the same root (sharing a

common “stem”) may be treated alike. This is appropriate in cases like “investment”, “investments”, “investor”, “investing” and “invest”, but Hull shows that “some form of stemming is almost always beneficial but the average absolute improvement due to stemming is small, ranging from 1 to 3%” (see [4]) The stemming can be done either by using the classic Porter’s algorithm (see [8]) or by a linguistic dictionary-based morphological analyser.

Significant improvements can be obtained in case of increased granularity when whole phrases rather than single words are used as indexing terms (e.g. “*Old Testament”*, rather than *Old* and *Testament*). The phrase list can be obtained either manually or automatically using a standard probabilistic algorithm. (see [3])

The application of different similarity measures between the document vectors is also a very important factor. Unfortunately, the limitations of the current paper do not permit us to illustrate and clarify in details the influence of each of these factors.

LSA in education

This section illustrates the application of LSA for analysis of textual information on an unordinary documents set: computer programs written in C. The experiments have been made among the first year Computer Science students from the Faculty of Mathematics and Informatics in the Sofia University “St. Kliment Ohridski”. During the semester the students have been offered 4 non-obligatory different algorithmic problems each of which had to be solved in 2 weeks. There have been 50, 47, 32 and 24 solutions for problems 1, 2, 3 and 4 respectively. As we expected, most of the solutions for each particular problem were very similar. This is obvious from figure 4, which shows the correlation matrices between the program solutions for each of the problems (in 20 dimensional space). This time we chose just 3 colors: white for 0%-50%, light gray for 50-90%, and black for 90%-100%. Except one case with problem 1, all other black cells corresponded to programs that have been copied one from the other with minor changes. In one particular case two of the programs for problem 2 were absolutely identical. Later we knew that two students have developed the program and each of them had submitted his copy with no modifications at all. In another case it was quite difficult to found an apparent similarity between two programs. But a deeper analysis showed that, although there have been made considerable manipulations on the source code the programs were identical. In all other cases we had no hesitations that the couples of similar programs found by LSA were in fact the same. What is interesting is the comparatively large amount of programs having correlation coefficient higher than 50%. This is normal because the programs written in C share the same reserved words (*for*, *if*, *while*, etc.) and the students are strongly limited in the ways to express themselves. So, it is unreasonable to claim that the gray cells show identical (copied) programs as well. The matrices with the corresponding correlation coefficients can be found on the Internet at: *http://www.comsoft.bg/preslav/pkurs/*.

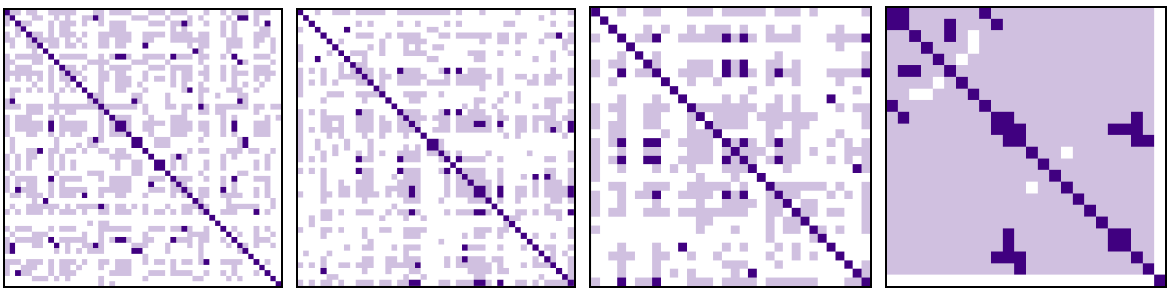

Figure 4. Correlation matrices for problems 1, 2, 3 and 4

Discussion

The purpose of this paper is to show the general application of LSA for textual information retrieval. Thus, the experiments above must be accepted with some

reserves. For example, the proximity between the English versions of two religious texts belonging to different religions whose original is not English is highly influenced by the translation process. This is especially the case with the religious texts that have been submitted to several translations before be published in English on the Internet. It is well known that the Egyptian Book of Dead was first translated to French and then from French to English.

Future work

We plan to analyze the proximity of the literature of the Bulgarian classic writers using LSA. We also plan to concentrate on some improvements of the general algorithm that will take in account the terms occurrence context.

References:

1. Berry M., Do T., O'Brien G., Krishna V., and Varadhan S., SVDPACKC (Version 1.0) User's Guide. April 1993.
2. Deerwester S., Dumais S., Furnas G., Laundauer T., Harshman R. Indexing by Latent Semantic Analysis. Journal of the American Society for Information Sciences, 41 (1990), pp. 391-47.
3. Dunning T., Accurate Methods for the Statistics of Surprise and Coincidence. Computational Linguistics, Volume 19, Number 1, 1993.
4. Hull, D. Stemming Algorithms: A case study for detailed evauation. In Journal of the American Society for Information Science. 47 (1):70-84, 1996.
5. Laudauer T., Foltz P., Laham D. Introduction to Latent Semantic Analysis. Discourse Processes, 25, pp. 259-284.
6. Landauer, T. K., Laham, D., Rehder, B., & Schreiner, M. E., (1997).How well can passage meaning be derived without using word order? A comparison of Latent Semantic Analysis and humans. In M. G. Shafto & P. Langley (Eds.), Proceedings of the 19th annual meeting of the Cognitive Science Society (pp. 412-417). Mawhwah, NJ: Erlbaum.
7. LSA 1990-99, see http://lsa.colorado.edu
8. Porter, M. An Algorithm for Suffix Stripping. Program, 14:130-137, 1980.
9. Religions, see http://davidwiley.com/religion.html
10. The Bible, see http://www.bible.org/netbible/download.htm
11. The Quran, see http://www.usc.edu/dept/MSA/quran/
12. Vlajic N., Card H. An adaptive Neural Network Approach to Hypertext Clustering. University of Manitoba. 1998.

For contacts:

Preslav Ivanov Nakov, Sofia University “St. Kliment Ohridski” and Rila Solutions
(+359 2) 97 97 309, GSM (+359 88) 373 609, preslav@rila.bg, preslav@rocketmail.com