The Use of Latent Semantic Indexing to Cluster Documents into their Subject Areas

Roseline Antai, Chris Fox, Udo Kruschwitz

University of Essex
Wivenhoe Park, Colchester, Essex,
United Kingdom
{rsanta, foxcj,udo}@essex.ac.uk

Abstract

Keyword matching information retrieval systems are plagued with problems of noise in the document collection, arising from synonymy and polysemy. This noise tends to hide the latent structure of the documents, hence reducing the accuracy of the information retrieval systems, as well as making it difficult for clustering algorithms to pick up on shared concepts, and effectively cluster similar documents. Latent Semantic Analysis (LSA) through its use of Singular Value Decomposition reduces the dimension of the document space, mapping it onto a smaller concept space devoid of this noise and making it easier to group similar documents together. This work is an exploratory report of the use of LSA to cluster a small dataset of documents according to their topic areas to see how LSA would fare in comparison to clustering with a clustering package, without LSA.

Keywords: Latent Semantic Indexing, Singular Value Decomposition, Information Retrieval, Document Clustering

1. Introduction

Would it not be just perfection if we could find all the relevant information we need in one place, without having to sieve through loads of irrelevant search results? This has been one of the goals of information retrieval researchers for years, and it still is. Information retrieval systems which utilize traditional search approaches, such as query term matching are often plagued by two problems – Polysemy and Synonymy (Dumais et al, 1988).

Dumais et al (1988) explain that synonymy arises from the fact that there exists such a huge diversity in the words used by people to explain/define the same object or concept. Now, these include the writers, as well as the searchers. In evidence to this, Deerwester et al (1990) gives the percentage of the instances which two people use the same major keyword for a certain object to be less than 20%. The cause for this may be due to different levels of education, geographical location, and backgrounds, to mention a few.

An example of synonymy can be seen in the interchangeable use of the words “cars”, “vehicles” and “automobiles” by some writers and searchers. A searcher using the search term “cars”, may not have results containing articles which use the term “vehicles” or “automobiles”, though they may be relevant. Same goes for the rest.

On Polysemy, Deerweester et al (1990) explained it as the condition in which words have more than one unique meaning. An example of polysemy can be the polysemous word “bank”, which can be used for a financial institution and a piece of land by water. Polysemy reduces the precision of an information retrieval system, and synonymy reduces its recall (Deerwester et al, 1990).

It is assumed that these problems cause ”noise” in the search space, and this noise obstructs the underlying latent structure which exists in the semantic space, leading to inadequate search results (Deerwester et al, 1990). Latent Sematic Analysis works on this assumption.

Having all relevant search items grouped into clusters improves the efficiency of the information retrieval system. According to Deerwester et al (1990), clustering empirically improves the computational efficiency of search.

1.1. Latent Semantic Analysis

LSA has been defined in different ways by different researchers. Hong (2000) defines LSA as a statistical information retrieval technique, designed for the purpose of reducing the problems of synonymy and polysemy in information retrieval. LSA is also defined as a technique used for automatic indexing and retrieval, which takes advantage of the semantic structure in correlating terms with documents, to improve the retrieval of documents relevant to a certain query. It was designed to solve the problem of retrieval methods which work by matching words in queries with words in documents (Deerwester et al, 1990). Wiemer-Hastings defines LSA to be a method used for comparing texts, using a vector-based representation, learned from the body of the documents. It is used to create vector-based representations of texts, which are claimed to capture the semantic content of such texts (Wiemer-Hastings, 1999).

LSA works by taking advantage of the conceptual content of documents, it makes no use of specific terms, but instead deals with the concept, and carries out a search on this basis (Hong, 2000).

The rationale behind it is that specific terms will be mapped onto this concept space, and other concepts can hence be retrieved from a particular concept, and through this retrieval, documents can be retrieved. This concept space is created using LSA’s Singular Value Decomposition (SVD)(Hong, 2000).

SVD is what distinguishes LSA from the more traditional Vector Space Model (VSM).VSM uses the original dimension of the document space, and this makes it much less effective than LSA, as does its use of the keyword

matching search technique. LSA works by using the term-document matrix, much like the Vector Space Model (VSM), but it improves on this traditional VSM through its dimension reduction process.
LSA works by first taking a document collection, and creating an association matrix of terms by documents, where the terms are placed on the rows, and documents on the columns, and the matrix entries are the frequencies with which each term appears in a corresponding document. The next step is pre-processing which involves stop word removal, and assignments of weights to terms. The third and most important is the Singular Value Decomposition (SVD).
SVD identified as the major strength of LSA, is defined by Dumais et al (1988) as a technique bearing a close resemblance to eigenvector decomposition and factor analysis, which takes a large matrix say, 'X', which is the association matrix of terms to documents, and then decomposes this matrix into a set of orthogonal factors, usually in the range of 50-150 factors, which can yield an approximate of the original matrix if linearly combined.
Paulsen and Ramampiaro (2009) summarize the role of SVD thus; SVD creates a semantic space from the original matrix, and decomposes it into the left and right singular vector matrices, and a diagonal matrix of singular values. The semantic space is made up of a term by concept space, which is the left singular vector matrix, the concept by document space, which is the right singular vector matrix, and the third matrix, the concept by concept space, which is the diagonal matrix.

Algebraically, this is represented thus:

$$\mathbf{X} \approx \hat{\mathbf{X}} = \mathbf{U}\mathbf{S}\mathbf{V}^{\mathbf{T}}$$

the original matrix 'X' is decomposed into 'U', ''S' and 'V[1]. The linear combination of the three will give rise to a matrix 'X^ ', which is a least square fit of matrix 'X'.
Another aspect of Latent Semantic Analysis which was also considered in this work was the choice of optimal dimensionality. The importance of this cannot be over-emphasized. Too large and noise will be let into the data, and too small, and some important concepts can be lost. We used the trial and error method similar to that used by Deerwester et al, (1990).

1.2. Clustering

One of the definitions given of clustering is that by Zaiane (1999), as a process in which a set of objects are split into a set of structured sub-classes, bearing a strong similarity to each other, such that they can be safely treated as a group. Such sub-classes are referred to as clusters. Csorba and Vajk (2006) define document clustering as a procedure which is used to divide documents based on a certain criterion, like topics, with the expectation that the clustering process should recognize these topics, and subsequently place the documents in the categories to which they belong.
There are various clustering algorithms, which work in different ways, and are named accordingly. The clustering algorithms we are concerned with are partitional clustering algorithms which discover clusters by performing a partitioning of the dataset into a number of clusters, automatically derived, or predetermined (Zhao and Karypis, 2001).
This work is concerned with clustering documents according to their subject areas, by first carrying out Latent Semantic Analysis to get rid of the noise and reduce the dimension of the semantic space which makes it easier for clustering algorithms to pick up on the shared concepts (Csorb and Vajk, 2006).
This work is organised as follows; Section 2 gives an overview of some related literature, in Section 3, the experimental procedure carried out in this work is explained, Section 4 gives the evaluation process, and Section 5 is the conclusion drawn from this work.

2. Related Work

LSA has been applied in quite a number of areas, some of which are mentioned here. LSA has been used in making matches between reviewers and papers they can review, based on other papers they liked (Landauer et al,1998), also in TREC [2,3,4], LSA recorded very good results, performing better than SMART and Telecordia's implementation (Hong, 2000). An LSA model was also used to simulate international student's performance in the TOEFL test, and this helped in arriving at the conclusion that the weight of the effect of word choice on the expression of meaning was greater than it was being credited for (Landauer et al,1998), LSA has also found applications in intelligent tutoring systems (by Graessar, 2000 and Wiemer-Hastings, 2004, as well as in flight simulators (Wiemer-Hastings, 1999).
Our work however is focused on LSA and clustering. Paulsen and Ramampiaro (2009) combine LSA with two different clustering algorithms. They create initial clusters based on LSA, and then implement the flat clustering method to perform a further grouping of similar documents in clusters. Their work was focused on improving the K-means algorithm, as it is less greedy. They used LSA in conjunction with flat clustering for the retrieval of biomedical information. Our work differs from this based on the fact that though we are using K-means for clustering as well, we are not focused on making K-means a less greedy algorithm, and we did not carry out a two-step clustering process, but applied LSA on the document set, and then the clustering algorithm on the reduced dimension set.
Some other work on document clustering/retrieval with LSA, have used document topics, or paragraphs as their document collection. Weimer-Hastings (1999), treated each paragraph as a different document, Deerwester et al (1990), worked on two document collections, where the documents were made up of the full text of the title and the abstract, Dumais et al (1988) and Landauer(1998) used document titles as their document collection, to mention but a few. In our work the document collection used was made up of the full text of the documents, and not only of abstracts and topics. Song and Park (2009) work on the development of a genetic algorithm method based on a latent semantic model (GAL) for text clustering. Using the Reuters-21578 dataset, and varying dimensions from 50 – 350, they arrived at the conclusion that the GALs which are the genetic algorithms using LSA outperform the GAs, which were the genetic algorithms using VSM (vector space model), and that GALs were also faster than GAs. They utilize the OpenCV software package for SVD.

A related but different approach to ours is that of Csorba and Vajk (2006), who perform double clustering in LSA in an attempt to solve problems posed by polysemy and synonymy. In their work, they used SVD to first reduce the dimension of the vector space, so as to carry out clustering in a space of lower dimensionality, and reduced noise. They then use the double clustering approach to further reduce noise in the vector space by carrying out a clustering of terms in the document corpus, and then carrying out the classification of documents in a feature space obtained from the term clusters. Our work differs from this as we do perform LSA on the document collection, using SVD to reduce the number of dimensions, and then carry out a single clustering on the vector space with reduced dimensions.

However, there are other approaches which are related to LSA. One of this is Probabilistic Latent Semantic Analysis (PLSA). PLSA is an approach to automated document indexing based on the statistical latent class model for factor analysis for count data (Hofman,1999). According to Hofman(1999), PLSA, which has a more solid statistical foundation, and defines a proper generative data model was developed to improve on LSA's lack of a solid statistical foundation, and failure to appropriately handle polysemous words.

Another related approach is the Latent Dirichlet Allocation (LDA), proposed by Blei et al(2003), which uses the fact that LSA and PLSA work by considering a document as a bag-of-words, and as such word order, as well as document order is not considered, and hence, this leads to an assumption of exchangeability, which in turn requires mixture models to capture this exchangeability. LDA is said to be a generative probabilistic model (more generative than PLSA) of a corpus, whose basic idea is that documents are represented as random mixtures over latent topics, and that each topic is characterized by a distribution over words (Blei et al., 2003).

3. Experiment

For our experiment, (see Fig 1) we used a document collection comprising 118 documents from four subject areas. The subject areas were deontics, evolutionary computing, semantics and imperatives. As a form of pre-processing, the documents were converted from PDF format, or any other format, to text format. The document collection was split into two equal halves, comprising each of 59 documents. One half was used as the training set, and the other was used as the test set.

The jLSI library was used to carry out our latent semantic analysis. Stop words were removed, and then SVD was performed in different dimensions ranging from 2 to 50 dimensions. This was performed first on the training set, where the matrix was reduced to different dimensions, starting at 2, and ending at 50. Each reduced matrix of a certain dimension was clustered, and the resulting clusters were evaluated. The test set was reduced to the dimension which gave the best clustering solution for the training set, and was subsequently, clustered. The trial and error method, and the notion of 'what works best' used by Deerwester et al (1990) was utilized in this work. The reason why we used this was because; we were dealing with a really small dataset, and also, we wanted

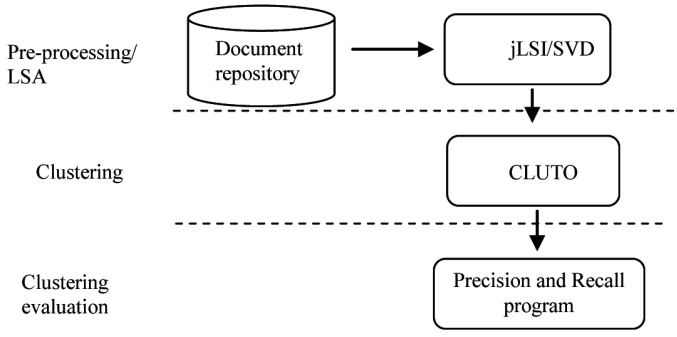

Fig. 1 : System Architecture

to see how the results fared at different dimensions, and to investigate the idea of an optimal dimensionality for a certain document size.

The K-means clustering algorithm, a type of partitional algorithm was used in this work. The decision to use this was due to its simplicity, and on the work carried out by Zhao and Karypis (2001), in which it was established that contrary to widespread belief of partitional algorithms being inferior to agglomerative ones, partitional clustering algorithms can yield better solutions and are therefore ideal for clustering large document sets, as they have low computational requirements, and yield higher quality clusters.

The clustering package used in this work was CLUTO. CLUTO is a clustering toolkit consisting of a suite of clustering algorithms, which are partitional and agglomerative, or hierarchical and non-hierarchical, based on the chosen scheme. CLUTO is used for clustering both high and low dimensional datasets, and also for analyzing the features of the different clusters (Karypis, 2003). CLUTO not only provides tools for analyzing the derived clusters, but also provides visualizations. Its partitional algorithms handle sparsity really well, should there be a case where the matrices are too sparse based on a really high value of dimensionality (Karypis, 2003). CLUTO uses two standalone programs to carry out clustering and provides analysis of the clustering results. These programs are the vcluster program and the scluster program. They cluster the data set into a predetermined number of clusters. The vcluster program's primary input is a matrix, which stores the objects to be clustered. Each row of the matrix represents a single object, and its various columns correspond to the dimensions (features) of the objects (Karypis, 2003). This format corresponds to the format of the matrix used in this experiment, and hence the vcluster program was used for this work.

The baseline used for this work was the clustering results produced by using CLUTO only, without first performing LSA. The document collection was converted straight into a matrix, using the doc2mat feature of CLUTO, preprocessing was carried out where stop words were removed. This was done using the same stop words list that was used for the LSA, and then the vcluster program was run on the matrix. The results of this are represented in Table 4 and 5.

4. Evaluation

Precision, recall and F-measure were used for the evaluation of the clustering results, and Table 1 gives their values for each cluster, cluster 0-3, obtained from using CLUTO without LSA. Table 2 gives the precision and recall values obtained from the clustering carried out on matrices of different dimensions, from 2 to 50, of LSA.

Looking at Tables 1 and 2, it appears that the baseline outperforms LSA. The only results comparable to the baseline are those obtained using the matrix reduced to 5-dimensions. The precision obtained from the baseline is higher than that obtained at 5-dimensions for cluster 0 and 2. The precision obtained at the 5-dimension is higher than that of the baseline for clusters 1 and 3. The opposite is the case for the recall values, with the baseline having better recall values for clusters 1 and 3, while LSA with 5-dimensions has better results for clusters 0 and 2.

Figure 2 also gives the visual plot of the average F-measure values for the four clusters obtained using LSA, over the different dimensions. Fluctuations of the F-measure values can be seen from these plots. It can also be seen that the F-measure value shoots up at the 5th dimension, and declines afterwards, reaching the lowest value on 50 dimensions.

Entropy and purity were the other two metrics used to evaluate the clustering results obtained in this work. CLUTO computes the entropy and purity of each cluster. These two metrics are used to measure the quality of clustering solutions. Entropy is concerned with the distribution of the different classes of documents within

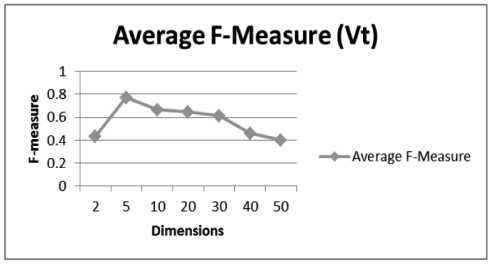

Figure 2. Average F-measure by dimensions plot

each cluster, while purity looks at the extent to which a particular cluster contains documents which are mainly from one class (Zhao and Karypis, 2003). The values of these two according to Karypis (2003) give an indication of the quality of the clustering solution, with low entropy values and high purity values indicating a good clustering solution. The results obtained from clustering with LSA at 5 dimensions had the lowest entropy value and highest purity result; it also had the highest precision values for each cluster, in comparison to the different dimensions,

CLUSTER	P	R	F-MEAS
0	0.89	0.85	0.87
1	0.86	1.0	0.92
2	0.83	0.63	0.72
3	0.64	0.82	0.72

Table 1. Baseline (Clustering with CLUTO only).

DIM	CLUSTER	P	R	F-MEAS
2	0	0.0	0.0	0.00
	1	0.38	0.38	0.38
	2	0.75	0.75	0.75
	3	0.48	0.83	0.61
5	0	0.71	1.0	0.83
	1	0.89	0.8	0.84
	2	0.69	0.82	0.75
	3	0.81	0.56	0.67
10	0	0.92	1.0	0.96
	1	0.41	0.82	0.55
	2	0.88	0.75	0.81
	3	0.57	0.25	0.35
20	0	0.89	0.73	0.80
	1	0.77	0.63	0.69
	2	0.67	0.4	0.50
	3	0.44	0.92	0.59
30	0	1.0	1.0	1.00
	1	0.45	0.82	0.58
	2	0.77	0.5	0.61
	3	0.29	0.25	0.27
40	0	0.47	0.5	0.48
	1	0.57	0.67	0.62
	2	0.6	0.3	0.40
	3	0.28	0.45	0.34
50	0	0.21	0.27	0.24
	1	0.4	0.38	0.39
	2	0.5	0.3	0.38
	3	0.5	0.75	0.60

Table 2. Clustering with LSA

2,10,20,30,40 and 50. Hence, this was also one of the reasons why this was selected as the best result.

The entropy value of the baseline was 0.407 which was lower than that obtained from clustering with LSA, which was 0.473. The purity value of the baseline was 0.814, which was higher than that of LSA with clustering, 0.780. From these values, and those of the precision and recall, it appeared that clustering without LSA gave better results. This raises the question, 'why bother with LSA?' A closer look though at the results presented in Table 3, 4, 5 and 6 actually shows that it seems LSA results actually do produce clusters with higher internal similarities. To explain this, a further analysis of the clustering result is necessary.

Cid	Size	**ISim**	Sem	Imp	Deo	Evo
0	17	**+0.731**	1	2	2	12
1	18	**+0.931**	0	16	2	0
2	13	**+0.811**	9	1	3	0
3	11	**+0.902**	1	1	9	0

Table 3. ISim from LSA vcluster result

Cid	Size	**ISim**	Sem	Imp	Deo	Evo
0	19	**+0.169**	1	17	1	0
1	14	**+0.159**	0	1	1	12
2	12	**+0.146**	1	1	10	0
3	14	**+0.136**	9	1	4	0

Table 4. ISim from CLUTO only vcluster result

Tables 3 and 4 show the cluster id or cluster number, the size of each cluster, 'ISim' value and the distribution of the objects in the different clusters. 'Deo', represents the deontics class, 'Sem' represents the semantics class, 'Evo' represents the Evolutionary computing class, and 'Imp' represents the Imperatives class.

'Cid' which gives the cluster number gives the order of the discovered clusters. Clusters that are tight and far away from the rest of the objects have smaller 'cid' values (Karypis,2003). From the tables, cluster 0 and 1 interchangeably have the smallest 'cid' value, but both are clusters of 'Evo' and 'Imp' documents, clustered by both LSA and CLUTO only, but still with the smallest 'cid' values.

'ISim' displays the average similarity between the objects of each cluster, that is, the internal similarities. (Karypis,2003). The values of 'ISim' for clustering without LSA as can be seen from Table 4 are quite low, in comparison to those of clustering with LSA in Table 3. For cluster 0, 'ISim' is +0.169, from Table 4, and +0.731 from Table 3, and for all the other clusters as well, the values of 'ISim' are much lower for the clustering results without LSA. These results tend to suggest that the clustering achieved using LSA gives clusters whose objects exhibit higher average similarities with each other, within the cluster, though the purity of the clusters without LSA is slightly higher than that with LSA.

The next step in the evaluation process was the analysis of the descriptive features of each cluster, shown in Tables 5 and 6. Five descriptive features are shown in both tables. The column numbers are only used here to represent the number of features, and not the features themselves. Hence, 'col 1' in cluster 0 does not represent the same feature as 'col 1' in cluster 1, and this goes for all the columns in the two tables. The percentage of within cluster similarity that a particular feature can explain is displayed by CLUTO beside the feature (five features by default). These are called the descriptive features. From the tables, it can be observed that these values are larger for the clustering result with LSA, than that of CLUTO only. As these percentages show how much of the similarity within clusters that a certain features can explain, it does seem as though the features obtained from clustering with LSA are features which hold more similarity information, than those obtained from the CLUTO only clustering.

Cluster 0					
Feature	Col 1	Col 2	Col 3	Col 4	Col 5
%	20.5	9.8	3.2	2.3	2.0
Cluster 1					
Feature	Col 1	Col 2	Col 3	Col 4	Col 5
%	9.2	6.4	4.0	3.8	3.3
Cluster 2					
Feature	Col 1	Col 2	Col 3	Col 4	Col 5
%	8.9	8.1	6.7	6.1	4.5
Cluster 3					
Feature	Col 1	Col 2	Col 3	Col 4	Col 5
%	9.1	3.4	3.2	2.5	2.4

Table 5. Descriptive features from CLUTO only vcluster result

Cluster 0					
Feature	Col 1	Col 2	Col 3	Col 4	Col 5
%	29.6	26.6	25.8	12.5	5.6
Cluster 1					
Feature	Col 1	Col 2	Col 3	Col 4	Col 5
%	44.6	42.7	10.5	2.0	0.3
Cluster 2					
Feature	Col 1	Col 2	Col 3	Col 4	Col 5
%	43.1	31.2	24.0	1.5	0.1
Cluster 3					
Feature	Col 1	Col 2	Col 3	Col 4	Col 5
%	38.6	26.3	17.7	17.4	0.0

Table 6. Descriptive features from LSA vcluster result

As a further evaluation, LSA was applied to the test set at a dimension of '5'. This was to test if a certain number of dimensions could lead to a good clustering result of document sets of the same size (an optimal dimension), and as was expected, the results obtained were worse off than those obtained with the training set.

5. Conclusion

Our aim was to cluster a set of documents according to their topic areas using LSA, we carried this out successfully and compared the results against a baseline, clustering with CLUTO, without LSA.

Though from the results obtained, it does seem that the internal cluster similarity is much higher when LSA is used, compared to when only clustering is carried out without LSA, and that the descriptive features produced when LSA is used give a higher percentage of within cluster similarity that a feature can explain, than when LSA is not used, it would be very ambitious to conclude that LSA does give better results, given the size of the data set.

We take into consideration the fact that the dataset was really small, and that clustering was carried out just once, hence a firm conclusion cannot be drawn from this.

What we present however, is an explorative report on the clustering with LSA, using a small dataset, to assess its performance, as it is more popularly used on larger datasets.

This work can be extended by using a larger document collection with more topics/document classes, to really get the best out of LSA.

References

Blei, D.M., Ng,A.Y., and Jordan, M.I.,(2003). Latent dirichlet allocation. *J. Mach. Learn. Res.* 3 (March 2003).

Csorba,K. and Vajk, I. (2006). Double Clustering in Latent Semantic Indexing. In Proceedings of SIAM, 4th Slovakan-Hungarian Joint Symposium on Applied Machine Intelligence, Herlany, Slovakia.

Deerwester, S., Dumais,S.T., Furnas,G., Landauer,T., Harshman,R. (1990). Indexing by Latent Semantic Analysis.Journal of the American Society for Information Science. Volume 41, issue 6, p.391-407.

Dumais,S.T., Furnas, G.W., Landauer,T.K., Deerwester, S., Harshman, R. (1988). Using Latent Semantic Analysis to improve access to textual information.In proceedings of the SIGCHI conference on human

factors in computing systems, p.281-285, Washington D.C, United States.

Hofmann,T (1999). Probabilistic latent semantic indexing. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval* (SIGIR '99). ACM, New York, NY, USA, 50-57.

Hong,J.(2000). Overview of Latent Semantic Indexing. Available [online] at : http://www.contentanalyst.com/images/images/overview_LSI.pdf . Last accessed on 30th September, 2010.

Karypis,G.,(2003). CLUTO* A Clustering toolkit.Release 2.1.1. Technical Report: #02-017.Department of Computer Science, University of Minneapolis.

Landauer,T.K., Foltz, P.W. and Laham,D. (1998). Introduction to Latent Semantic Analysis.Discourse Processes, 25, 259-284.

Paulsen J.R. and Ramampiaro, H. (2009). Combining latent semantic indexing and clustering to retrieve and cluster biomedical information: A 2-step approach, NorskInformatikonferanse (NIK), 2009.

Song W. And Park S.C. (2009). Genetic algorithm for text clustering based on latent semantic indexing. Comput. Math. Appl. 57, 11-12 (June 2009), 1901-1907.

Wiemer-Hastings, P.(1999). Latent Semantic Analysis.In proceedings of the sixteenth International Joint conference on artificial intelligence. Volume 16, Number 2, p. 932-941.

Zaiane, O. (1999). Principles of Knowledge Discovery in databases, chapter 8: Data Clustering, lecturing slides for CmPUT 690, University of Alberta. Available [online]at: http://www.cs.ualberta.ca/~zaiane/courses/cmput690/slides/chapter 8/.

Zhao, Y and Karypis G. (2001). Criterion Functions for Document Clustering: Experiments and Analysis. Technical Report #01-40, Department of Computer Science, University of Minneapolis.