Tag Cloud Control
by Latent Semantic Analysis

submitted by
Angelina Velinska

supervised by
Prof. Dr. Ralf Möller
Dipl. Ing. Sylvia Melzer
Software Systems Institute (STS)
Technical University of Hamburg-Harburg

Dr. Michael Fritsch
CoreMedia AG
Hamburg

Contents

List of Figures

Chapter 1

Introduction

1.1 Motivation

Information Retrieval (IR) systems become more important not only due to their use in Internet and digital libraries, but also because the majority of companies organize their activities and depend on digital documents, and information. Finding the right information brings value to the business, and failing to do so usually leads to losses.

Certain factors influence the performance of search applications. On one side is the constantly growing problem of information overload. On the other, it is the peculiarities of humans users. IR systems need to adapt to their users, and to their information need.

- **Human behavior**. Users searching for information usually submit queries composed of a few keywords only, as shown in studies [1]. The search application performs exact matching between the submitted keywords, and the document set it searches through, and often returns a long list of results. When searching with short queries, the focus of a user is unclear, and the missing information contributes to long result lists containing many unrelevant hits. Users without domain expertise are not familiar with the appropriate terminology thus not submitting the right query terms with respect to relevance or specialization. Another issue is the ambiguity of words, when words have more than one meaning. As a consequence, search results do not fit the information needs of users. When relevant documents contain words that are semantically relevant to the queries but not the same (synonyms), they will not be judged relevant.
- **Manual processing of results**. When a large number of match-

ing documents is returned as a search result, only some of the documents can be read, due to human limitations in information processing, and time constraints (users want to find information quickly). Human users need to narrow down the search iteratively by reformulating the query, since it is unclear in which context their queried words are used, and which words could be useful to focus the search. This reformulation is known to be a frustrating and time consuming process.

- **Information need**. Search applications that implement the keyword search paradigm (or full-text search) are broadly accepted by users; however, the challenge for the next years is the development of search solutions that reflect users' context ("what the user meant" versus "what the user entered as a query"). In other words, solutions that are able to: *a*) organize search results better than in the form of long lists, *b*) adapt to a users personal skills and experience concerning the underlying document collection, and *c*) adapt to the retrieval task a user is concerned with; in other words, to adapt to the users' information need.

The factors listed above have lead to the development of techniques that assist users to effectively navigate and organize search results, with the goal to find the documents best matching their needs. *Document clustering* is a process of forming groups (clusters) of similar objects from a given set of inputs. When applied to search results, clustering organizes the results into a number of easily browsable thematic groups. It contributes to solving the problem of finding a meaningful ordering in a large amount of returned results. *Latent Semantic Analysis* is another method from IR that handles the problem of words having more than one meaning, or words ambiguity. It also filters out noise well, and reduced the number of unrelevant hits. Both techniques have been implemented in this work.

After documents have been clustered into categories according to their topics, they have to be presented to the users. In particular, the categories have to be labeled with characteristic terms for browsing. Which words from the cluster to choose as labels is a difficult problem to solve. This work presents algorithms for cluster labeling. It evaluates an interesting new algorithm, called Weighted Centroid Covering (WCC), proposed in [2] and [3]. Further, it proposes an improvement of WCC by using external semantic knowledge for definition of the cluster labels, provided by a domain-specific ontology, which was developed as a part of this work.

1.2 Information Retrieval systems

As a part of this work, we have implemented techniques from the field of IR. Therefore, what follows is an overview of the text analysis processes, common to most IR systems. Then, in the context of these processes, we shall summarize the contributions presented in this thesis.

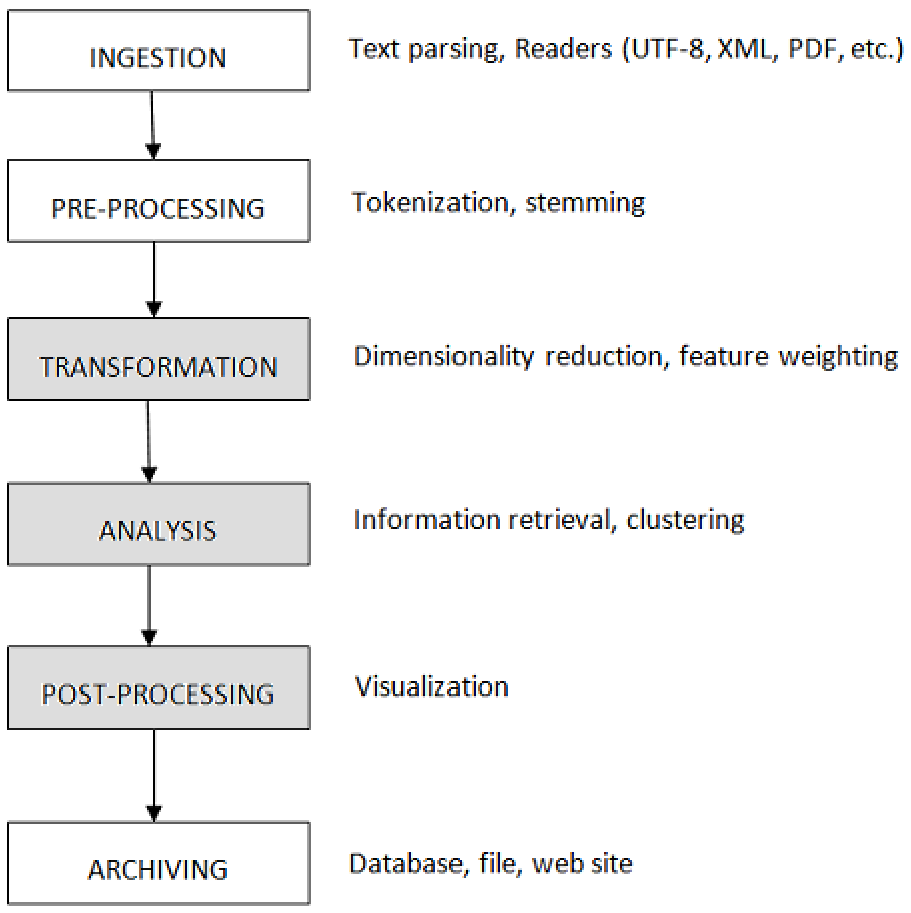

Figure 1.1: Workflow in IR systems

Most IR systems share common workflow, and follow common data processing stages. Some of the processes involved in text analysis are: lexical analysis to study word frequency distributions of words, retrieval, tagging/annotation, and visualization among the others. The workflow in IR systems usually starts with **document ingestion**, where texts from the document corpus[1], which will be analyzed, are parsed or loaded into memory. After that comes the **pre-processing** phase, responsible for text extraction, tokenization, token-filtering, text folding, etc. In this phase, documents are often represented as vectors, whose components quantify properties like how often words occur in the document. In the **transformation** phase is where a dictionary, matrix or other form of index is created from the extracted texts. In this phase all extracted terms[2] are assigned weights by applying a weight function. In Chapter 2.5 are given more details about the most common weight functions used in IR systems. Phase **analysis** can include retrieval process by querying the corpus, clustering, etc. **Visualization** phase is where concepts, query results, or summarizations extracted from the text collection are presented to users. And in the final **archiving** phase, results from the text analysis process can be saved.

The workflow in fig. 1.1 doesn't show any iterations or cycles between phases for simplicity. Iterations and repetition of certain phases of analysis, however, are common, e.g. between transformation and post-processing phases, or analysis and post-processing.

The focus of the current work is on transformation, analysis and post-processing phases of IR workflow.

1.3 Goal and scope of work

This works contributes with the following:

1. It offers an overview of the current cluster labeling algorithms, and makes an evaluation of WCC, proposed for unsupervised topic labeling of clusters in [2] and [3].

2. It proposes an improvement in WCC algorithm, performing topic identification based on external knowledge. A light-weight ontology has been developed for this purpose, in order to be used as a reference for external semantic knowledge during cluster labeling.

3. A software application for executing an IR process has been developed. It implements Latent Semantic Analysis (LSA) for information retrieval(analysis phase from fig. 1.1, and WCC for visualization of the main concepts contained in a document set in the form

[1]Throughout this work we use *document corpus* as a synonym to a document collection, in which IR tasks are performed.

[2]A term in this context denotes a word after pre-processing of the texts. A term is a single unit, e.g. a word, a phrase, a number, or an abbreviation.

of a tag cloud.

1.4 Outline

This chapter motivates the presented research work, and summarizes its contributions. It also offers a general overview to the phases of text analysis, common to most IR systems. Chapter 2 gives theoretical foundations for preprocessing and transformation phases of text analysis, and reviews a specific technique for information retrieval, called Latent Semantic Analysis. Chapter 3 contains the contribution related to evaluation of WCC algorithm, and a proposal for its improvement, by using a light-weight domain ontology. It is the analytical phase of the IR process. Chapter 4 refers to the post-processing phase of text analysis, giving the visualization means for presenting the main concepts retrieved from a document set. The software contribution, developed as a part of this work, is described in Chapter 5. Finally, in Chapter 6 the thesis concludes with evaluation of results and outlook.

Chapter 2

Latent Semantic Analysis

Summary. *The chapter gives a theoretical overview of LSA in the context of its use in this work.*

2.1 Overview

LSA was first introduced in [4] and [5] as a technique for improving information retrieval. Most search engines work by matching words in a user's query with words in documents. Such information retrieval systems that depend on lexical matching have to deal with two problems: synonymy and polysemy. Due to the many meanings which the same word can have, also called polysemy, irrelevant information is retrieved when searching. And as there are different ways to describe the same concept, or synonymy, important information can be missed. LSA has been proposed to address these fundamental retrieval problems, having as a key idea dimension reduction technique, which maps documents and terms into a lower dimensional semantic space. LSA models the relationships among documents based on their constituent words, and the relationships between words based on their occurrence in documents. By using fewer dimensions than there are unique words, LSA induces similarities among words including ones that have never occurred together [6]. There are three basic steps in using LSA: text pre-processing, computing Singular Value Decomposition (SVD) and dimensionality reduction, and querying the constructed semantic space.

2.2 Text pre-processing

If we have a document collection or a text corpus, on which we want to apply LSA, the initial step is to pre-process the texts into a suitable form for running LSA. Pre-processing can include a number of techniques, depending on the application requirements. The process of parsing, also called tokenization, is breaking the input text stream into useable tokens. During tokenization, filtering can be applied, i.e. removing HTML tags or other markup, as well as stop-wording, and removing punctuation marks. Stop words don't convey information specific to the text corpus, but occur frequently, such as: $a, an, and, any, some, that, this, to$.

A distinction has to be made between words or terms, and tokens. A term is the class which is used as a unit during parsing, and a token is each occurence of this class. For example, in the sentence:

> *CoreMedia CMS is shipped with an installation program for interactive graphical installation and configuration of the software.*

the term *installation* is represented by two tokens.

There is no universal way in which to parse a text, and the parsing decisions to address depend on the application in which the text collection will be used. Text parsing will influence all posterior processing in the following stages of LSA.

After tokenization, one has to construct a term-document matrix eq. (2.1). Having as rows the terms, and as columns the documents, its elements are the occurrences of each term in a particular document, where a_{ij} denotes the frequency with which term i occurs in document j. The size of the matrix is **m x n**, where **m** is the number of terms, and **n** is the number of documents in the text collection. Since every term doesn't appear in each document, the matrix is usually sparse.

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix} \tag{2.1}$$

Local and global weightings are applied to increase or decrease the importance of terms within documents. We can write

$$a_{ij} = L(i,j) \times G(i), \tag{2.2}$$

where $L(i, j)$ is the local weighting of the term i in document j, and $G(i)$ is the global weighting for term i. The choice of a weight function has impact on LSA performance, therefore in Section 2.5 we give an overview of the most common weight functions.

2.3 Singular Value Decomposition

After the initial pre-processing, the term-document matrix is decomposed into three matrices eq. (2.3) by applying Singular Value Decomposition (SVD). It is a unique decomposition of a matrix into the product of three matrices - U and V are ortonormal matrices, and Σ is a diagonal matrix having singular values on its diagonal.

$$A = U\Sigma V^T \tag{2.3}$$

After the initial matrix A is decomposed, all but the highest k valued of S are set to 0. The resulting reduced matrix is the semantic space of the text collection. A classical example presenting the truncated SVD [4] can be used for displaying dimensionality reduction, and how it affects all three matrices.

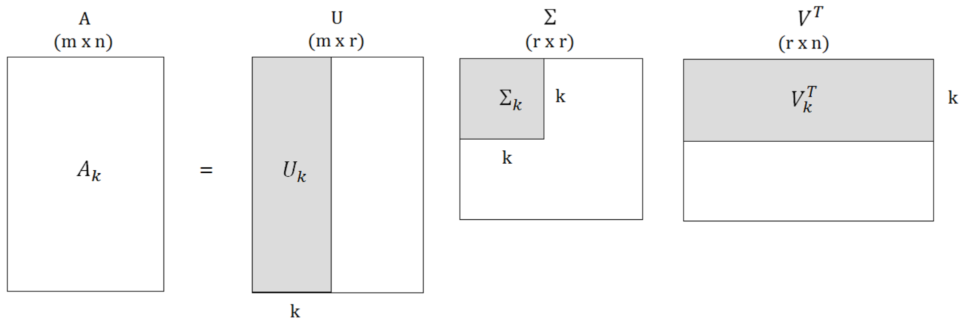

Figure 2.1: Diagram of truncated SVD

A_k - best rank-k approximation of A	m - number of terms
U - term vectors	n - number of documents
Σ - singular values	k - number of factors
V^T - document vectors	r - rank of A

Figure 2.1 is a visual representation of SVD as defined in equation (2.3). U and V are considered as containing the term and document vectors

respectively, and Σ is constructed by the singular values of A. An imporant property of SVD is that the singular values placed on the diagonal of Σ are in decreasing order. Hence, if all but the first k singular values are set to 0, the semantic meaning in the resulting space is preserved to some approximation k, while noise or variability in word usage, is filtered out. Noise in this case are the terms with lowest weights which carry little meaning. By using fewer dimensions k, LSA induces similarities amont terms including ones that have never occured together. Terms which occur in similar documents, for example, will be near each other in the k-dimensional space even if they never co-occur in the same document. This means that some documents which do not share any words with a users query may be near it in k-space.

A factor to be considered when computing SVD is the run-time complexity of the algorithm. For decomposition of very large matrices, it is $O(n^2k^3)$, where n is the number of terms in the text corpus, and k is the number of dimensions in semantic space after dimensionality reduction. Note that k is typically a small number between 50 and 350.

A more detailed description of SVD can be found in [7] and [8].

2.4 Querying the semantic space

In this work we are using LSA for IR purpose. Therefore, the final step of applying the technique is to pose queries on the constructed semantic space. A query q is a set of words which must be represented as a document in the k-dimensional space, in order to be compared to other documents. The user's query can be represented by

$$q = q^T U_k \Sigma_k^{-1} \tag{2.4}$$

where q is the set of words in the query, multiplied by the reduced term and singular values matrices. Using the transformation in eq. (2.4), the query is "mapped" onto the reduced k-space. After the mapping, the resulting query vector can be compared to the documents in the k-space, and the results ranked by their similarity or nearness to the query. A common similarity measure is the cosine between the query and the document vector. From the resulting document set, the documents closest to the query above certain treshold are returned.

2.5 Factors influencing LSA performance

The effective usage of LSA is a process of a sophisticated tuning. Several factors can influence the performance of the technique. These factors are pre-processing of texts (removal of stop-words, filtering, stemming), frequency matrix transformations, choice of dimensionality k, choice of similarity measure.

Dumais et al. [9] and Nakov et al. [10] have carried research on LSA performance depending on the choice of factors such as frequency matrix transformations, similarity measures, and choice of dimension reduction parameter k. They conclude that performance based on the choice of these factors depends on the particular text corpus, as well as on the purpose of LSA application. However, in the case of matrix transform, log-entropy performs better as compared to other matrix transform function combinations, including the popular term frequency - inverse document frequency ($tf \times idf$). Therefore, we implement the former in this work.

Local function: logarithm	$L(i,j) = \log(tf(i,j) + 1)$
Global function: entropy	$G(i) = 1 + \frac{\Sigma_j p(i,j)}{\log n}$

where n is the number of documents in the collection.

Further, it has been stated ([9],[11]) that with respect to similarity measures used, LSA performs optimal when cosine similarity measure is implemented to calculate the distance between vectors in the semantic space. We have therefore used it to measure the relevance between queries and documents. The cosine measure between two vectors d_1 and d_2 is given by:

$$sim(d1, d2) = \frac{\overrightarrow{V}(d_1).\overrightarrow{V}(d_2)}{\left|\overrightarrow{V}(d_1)\right| . \left|\overrightarrow{V}(d_2)\right|} \tag{2.5}$$

Dimensionality reduction parameter k is defined empirically based on the experimentation results presented in Chapter **??**.

Acronyms

IR	Information Retrieval.
LSA	Latent Semantic Analysis.
SVD	Singular Value Decomposition.
WCC	Weighted Centroid Covering.

Appendix A

Appendix

TODO: insert important source code parts here.

Bibliography

[1] A. Spink, J. Bateman, and B. J. Jansen, "Users' searching behavior on the excite web search engine," in *WebNet*, 1998.

[2] B. Stein and S. M. Z. Eissen, "Topic identification: Framework and application," in *Proc of International Conference on Knowledge Management (I-KNOW)*, 2004.

[3] B. Stein and S. M. zu Eissen, "Topic identification," *KI*, vol. 21, no. 3, pp. 16–22, 2007.

[4] S. T. Dumais, G. W. Furnas, T. K. Landauer, S. Deerwester, and R. Harshman, "Using Latent Semantic Analysis to improve access to textual information," in *Sigchi Conference on Human Factors in Computing Systems*, pp. 281–285, ACM, 1988.

[5] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman, "Indexing by Latent Semantic Analysis," *Journal of the American Society for Information Science*, vol. 41, pp. 391–407, 1990.

[6] S. Dumais, "LSA and Information Retrieval: Getting Back to Basics," pp. 293–321, 2007.

[7] M. W. Berry, S. Dumais, G. O'Brien, M. W. Berry, S. T. Dumais, and Gavin, "Using linear algebra for intelligent information retrieval," *SIAM Review*, vol. 37, pp. 573–595, 1995.

[8] G. H. Golub and C. F. Van Loan, *Matrix computations (3rd ed.)*. Baltimore, MD, USA: Johns Hopkins University Press, 1996.

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

[10] P. Nakov, A. Popova, and P. Mateev, "Weight functions impact on LSA performance," in *EuroConference RANLP'2001 (Recent Advances in NLP*, pp. 187–193, 2001.

[11] P. Nakov, "Getting better results with Latent Semantic Indexing," in *In Proceedings of the Students Prenetations at ESSLLI-2000*, pp. 156–166, 2000.