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

Declaration

I declare that:
this work has been prepared by myself,
all literal or content based quotations are clearly pointed out,
and no other sources or aids than the declared ones have been used.

Hamburg, October, 2010
Angelina Velinska

Acknowledgements

TO BE DONE

Contents

List of Figures

Chapter 1

Introduction

Identifying the main concepts in texts is the subject of many research studies in the field of information retrieval and data mining.

This work investigates the implementation of Latent Semantic Analysis (LSA) for discovering the main concepts in texts, in order to present an overview of the text content in the form of a tag cloud.

1. introductory words, why is this work being written
2. mention information retrieval, lsa, tag clouds - generally
3. mention cms ? document collections ? content ?
4. mention the work of david mugo

During the last decade there have been constant optimizations in information retrieval effectiveness, making web search the preferred source of finding information. A substantial part of information retrieval deals with providing access to unstructured information in various domains. Information Retrieval (IR) refers to finding material (usually documents) of an unstructured nature (usually text) that satisfies an information need from within large collections (usually stored on computers) [1]. Many people today use methods from the field of IR when they use a search engine online, or search through their emails. In this context ”unstructured data” refers to data which does not have a clear structure.

IR technologies find wide application - in search engines, for browsing or filtering document collections, for further processing a set of retrieved documents. Before retrieval the documents are indexed, otherwise at each search, they would have to be scanned through for each query. The index maps the words or terms back to the documents where they occur. A method for document indexing, which is applied in this work, is called Latent Semantic Analysis (LSA). It indexes the document collection by representing it as a reduced matrix of words and documents. LSA representation improves IR performance with respect to a basic problem of word-matching search - synonymy, or the case when more than one term describe the same concept.

While IR deals with retrieval of documents, other systems manage content, such as documents. Content management includes a set of technologies and processes that support the creation, management and publication of content in any form or medium. Content may be documents, multi-media files, or any other file types that follow content lifecycle and require management. Content Management Systems (CMS) vary depending on their purpose and target environments - there are CMS for the web, for enterprise, for mobile devices, as well as CMS for managing collection of documents.

1.1 Motivation and objective

A drawback of the classical LSA implementation as an IR method is the low precision of the returned results. A previous work by David Mugo [2] has investigated the improvement of LSA precision performance by annotating the document collection and including the anotations used in LSA. In his work, Mugo constructs a concept-document matrix from the annotations used, and concatenates it with the word-document matrix normally generated in LSA process. The proposed solution, however, results in a slow speed of LSA, and has left Mugo's hypothesis open.

Taking into consideration the results from Mugo's work, the current project has several objectives to reach. It will investigate the implementation of LSA method for improving information retrieval in a domain-specific document management system with respect to context-based search. A further investigation will be made on improving the precision performance of LSA method by using semantic annotations, and on finding an adequate way to present the results of LSA as a tag cloud. And finally, it will be investigated how to use the tag cloud as a form of a relevance feedback to control LSA method.

In the context of the stated objectives, semantic annotations are meta data annotations used to add information to unstructured data, or to the document collection. Semantic annotations are based on an ontology in our case, specifically developed for the domain of interest CoreMedia CMS. Ontologies are used to capture some knowledge about a certain domain, by describing the concepts of the domain and the relationships between them. To further clarify the objectives, relevance feedback is an IR technique, used to influence the retrieved results based on the user's preference. It allows the user to modify the initial tag cloud by selecting the most relevant words. The tag cloud is then re-generated from LSA results with the relevance feedback posted as a query.

1.2 Outline

The reminder of this work is organized as follows. Chapter **??** describes in more detail what a document management system is, and provides an overview of the general structure of DocMachine 2.0, the CMS deployed at CoreMedia AG. Chapter **??** presents the basic concepts of ontologies and document annotations based on ontologies. In Chapter 2 an overview of latent semantic analysis method is given, as well as an approach for improving LSA's precision by including semantic annotations in the method. Chapter **??** presents the prototype implementation and makes an evaluation of the results achieved in this work. And finally, conclusions are drawn in Chapter **??**, along with some limitations of the current study and outlook for a future research.

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 [3] and [4] 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 that there are unique words, LSA induces similarities among words including ones that have never occurred together [5]. There are three basic steps to using LSA: parsing text, computing a Singular Value Decomposition (SVD), and mapping queries on the generated semantic space, so that similarities between documents, or queries and documents can be computed.

2.2 Text pre-processing

Before applying LSA, some pre-processing to the texts in the document collection is required. 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 words and punctuation marks. Stop words are such words that don't hold useful information but occur frequently in the texts, and therefore it is sensible to be removed. Examples of stop words are: $a, an, and, any, some, that, this, to.$

An important 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. All posterior processing in the following stages of LSA will be determined by the parsing.

After tokenization, one has to comupute a term - document matrix. Having as rows the terms, and as columns the documents, its elements are number of occurrence of each word in each document. The matrix is sparse, as not all terms occur in all documents.

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

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. Then, the term occurences are transformed into weights using a weight function, such as entropy, term frequency, inverse document frequency. These weights are represented by entries in matrix 2.1, where a_{ij} gives the weight of term i in document j.

2.3 Singular Value Decomposition

After the initial pre-processing, the generated term-document matrix is decomposed into three matrices by a process called 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 S is a diagonal matrix with singular values on its diagonal, as in 2.2.

$$A = USV^T \tag{2.2}$$

After the initial matrix A is decomposed by SVD, 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 from [3] presenting the truncated SVD can be used for displaying dimensionality reduction, and how it affects all three matrices (Figure 2.1).

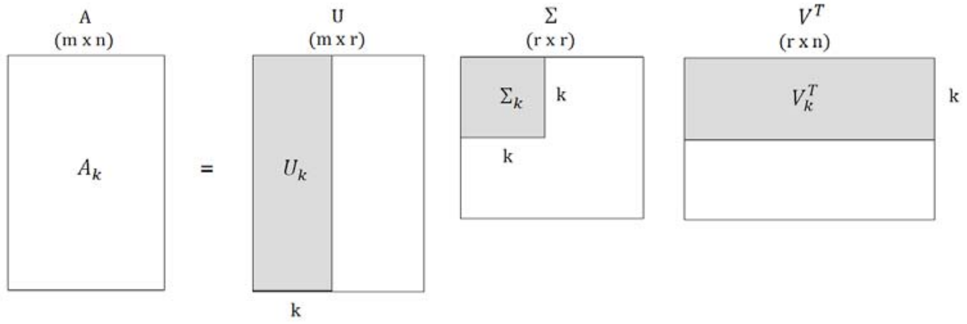

Figure 2.1: Diagram of the truncated SVD

LSA models the relationships between documents based on the words they contain, and the relationships between words based on their occurrence in the documents. By using fewer dimensions that there are unique terms, LSA induces similarities among terms including ones that have never occurred together [5]. SVD is also a tool for dimensionality reduction, and hence also noise reduction, as it sets to 0 the lowest term weights.

For decomposition of very large matrices, it is important to note that the run time complexity for performing SVD is $O(n^2k^3)$, where n is the number of terms, and k is the number of dimensions in semantic space after dimensionality reduction. k is typically a small number between 50 and 350.

For more detailed information on SVD, please refer to [6] and [7].

2.4 Query mapping in the semantic space

The final step of applying LSA is to compute similarities between entities in the semantic space. This includes computing similarities between queries and documents in the semantic space. In this case the query q has to be ”translated” as a document from the semantic space, by using 2.3.q^T is the transposed query vector, U_k is the reduced term matrix, and S_k^{-1} is the reduced singular values matrix.

$$q = q^T U_k S_k^{-1} \tag{2.3}$$

Search in the reduced vector space after SVD is done based on a similarity measure and coocurence of the terms within the documents. The decomposition finds the optimal projection into low-dimensional vector space.

2.5 Factors influencing LSA performance

Several factors influence the quality of results which LSA delivers. These factors are pre-processing (removal of stop-words, stemming, lemmatization), frequency matrix transformations, choice of dimensionality, choice of similarity measure.
A study by Nakov, Popova, Mateev[8] has summarized the influence of those factors on LSA, and has concluded that...

TODO:here give a formula for weighting function, and for similarity measure. Explain citing Nakov’s paper why you are using exactly these.

There are three main factors that can influence the performance of LSA[8][9]:

- Frequency matrix transoformations (choice of weighting function)
- Choice of dimensionality
- Text preprocessing prior to SVD, choice of similarity measure (???)

Further, the choice of dimensionality is dependent upon the matrix transformations performed, as pointed out by Nakov[9].

2.6 Advantages and drawbacks

why am i using lsa instead of lda for example?

1. PLSA - characteristics, advantages, disadvantages
2. LDA - characteristics, advantages, disadvantages

2.7 Latest development in the field of LSA

LSAView is a tool for visual exploration of latent semantic modelling, developed at Sandia National Laboratories [10].

at the end- improvements of lsa with the basics explained.

Acronyms

CMS Content Management Systems.

IR Information Retrieval.

LSA Latent Semantic Analysis.

SVD Singular Value Decomposition.

Appendix A

Appendix

You should not print the full source code :-). But note that the chapters are now called “appendix” and numbered with letters.

Bibliography

[1] C. D. Manning, P. Raghavan, and H. Schtze, *Introduction to Information Retrieval*. New York, NY, USA: Cambridge University Press, 2008.

[2] D. M. Mugo, "Connecting people using Latent Semantic Analysis for knowledge sharing," Master's thesis, Hamburg University of Technology, Jan. 2010.

[3] 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.

[4] 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.

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

[6] 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.

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

[8] 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.

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

[10] P. Crossno, D. Dunlavy, and T. Shead, "Lsaview: A tool for visual exploration of Latent Semantic Modeling," in *IEEE Symposium on Visual Analytics Science and Technology*, 2009.