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, December 2010
Angelina Velinska

Acknowledgements

I would like to thank Prof. Dr. Ralf Möller for his advice and guidance throughout the work on this thesis.
My special gratitude goes to Dr. Michael Fritsch in CoreMedia AG for his great patience, and constant support.

Contents

List of Figures

Chapter 1

Introduction

We are now in the years before the Semantic Web, or Web 3.0. In the past years there has been major research in the fields of Information Retrieval, and improving search systems. Major companies like Google offer already products implementing techniques from the field of IR or semantics, in order to improve precision of search results, and increase user satisfaction. However, despite the major research done in the field, the implementation of these techniques is still relatively limited on a business scale.

While many web-based search engines and applications implement techniques and tools from IR and semantic web, search functionality implemented in companies still relies mostly on full-text based search.

In order to create automatic modeling, processing, and analysis of unstructured text (LSA)

and labeling, and classification of unstructured text(Clustering and topic identification), we investigate the implementation of IR techniques for business needs in CoreMedia AG.

1.1 Motivation

Full-text search delivers a large number of results and does not handle synonymy and polysemy well. This problem can be partially solved by using IR technique to cope with polysemy and synonymy (such as LSA), and to use clustering for categorization of search results.
However, when using clustering, another issue arises - how to automatically identify category labels.

1.2 Goal and scope of this work

The goal of this work is to investigate implementation of IR technique LSA for handling ambiguous search in the use case of CoreMedia AG, Hamburg. In order to categorize search results, and provide better user experience, categorization of search results has been proposed, and a cluster labeling algorithm investigated. the implementation of IR and semantic technologies for business needs, in order to improve user experience with search systems.
Investigations have been made in three topics: use of ontologies to add semantic meaning to unstructured texts, use of IR techniques for document retrieval (LSA), and use of clustering technique and topic identification algorithm for categorizing search results.

1.3 Thesis Structure

Text analysis: methods for
searching - IR
labeling - TI algo
analyzing document collections - LSA, clustering

challenges:
- too much information to process manually (need automation) data ambiguity

- ambiguous queries lead to information overload and topic confusion (IR)
- determine topics in text collections and identify most important relationships (Topic detection and association); clustering and visualization and key analysis methods

active text analysis research, implementation of techniques in business environment, for business needs?

——————————-

——————————-

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.

1.4 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.

1.5 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 4 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 5 presents the prototype implementation and makes an evaluation of the results achieved in this work. And finally, conclusions are drawn in Chapter 6, 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: 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 (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 (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 [3] can be used for displaying dimensionality reduction, and how it affects all three matrices.

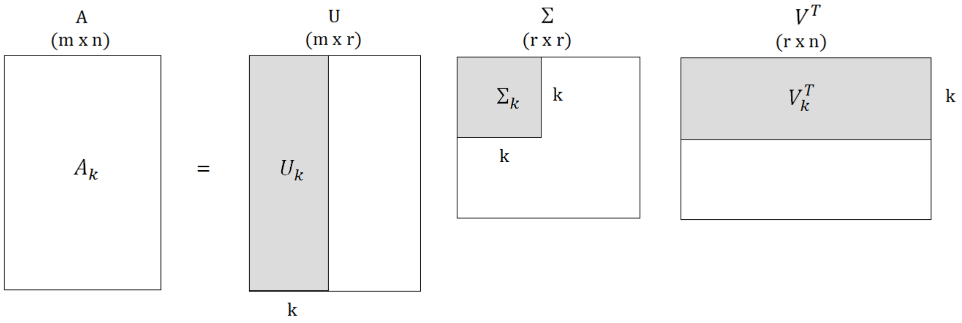


Figure 2.1: Diagram of truncated SVD

A_k - best rank-k approximation of A
U - term vectors
Σ - singular values
V^T - document vectors

m - number of terms
n - number of documents
k - number of factors
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 [6] and [7].

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 (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. [8] and Nakov et al. [9] 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 ([8],[10]) 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 5.

Chapter 3

Topic identification in clusters

A major problem in text analysis is to determine the topics in a text collections and identify the most important, novel or significant relationships between topics. Clustering and visualizations (tag clouds) are key analysis methods in order to solve this problem.

Clustering is widely used for recommendation, and for categorizing search. An example of a recommendation is "X" like these. The search engine will look for similar results as the ones presented.

disadvantages of clustering:
objects can be assigned to one cluster only
in social networks, clustering can be used to recognize communities in large groups of people.
clustering is also used in partitioning web documents into groups, a.k.a. genres (data mining)
search engines - categorization of search results or grouping (Yippy search engine)
recommender systems - recommend new items based on user's taste

When performing classification by clustering, the cluster labels are usually manually created by human beings. However, this is a very expensive approach. It is sensible to find and algorithm for automatically identifying topic labels or cluster labels. Therefore, we have investigated the performance of Topic identification algorithm by Stein and zu Eissen[11].

3.1 External topic identification

The best scenario is that cluster labels should present a conceptualization of the documents in text corpus. This is not achieved by the algorithm presented. Technically, a hierarchical clustering algorithm can construct from each Document set D a category tree. However, the labeling based on this hierarchical clustering will be far from a semantical taxonomy. This weakness of the algorithm presented can be corrected by using an external classification knowledge, e.g. an upper-level ontology.

We believe that the weaknesses of topic identification algorithms in categorizing search engines could be overcome if external classification knowledge were brought in. We now outline the ideas of such an approach where both topic descriptors and hierarchy information from an upper ontology are utilized.

Then, topic identification is based on the following paradigms:
1. Initially, no hierarchy (refines-relation) is presumed among the C 2 C. This is in accordance with the observations made in [Ertz et al. 2001].
2. Each category C 2 C is associated to its most similar set O 2 O. If the association is unique, $To(O)$ is selected as category label for C.
3. Categories which cannot be associated uniquely within O are treated by a polythetic, equivalence-presuming labeling strategy in a standard way. In essence, finding a labeling for a categorization C using an ontology O means to construct a hierarchical classifier, since one has to map the centroid vectors of the clusters C 2 C onto the best-matching O 2 O. Note that a variety of machine learning techniques has successfully been applied to this problem; they include Bayesian classifiers, SVMs, decision trees, neural networks, regression techniques, and nearest neighbor classifiers.

Chapter 4

Tag Clouds

We have found so far no other application implementing LSA in order to present an overview of the main topics found in a collection of unstructured texts, based on user queries. The TagCloud Summarizer is in this sense new.

There exist, however, search engines, which utilize categorizing of search results, such as Yippy(former Clusty, Vivisimo). It utilizes search, classification, and Social web (Web 2.0).

4.1 existing implementations

http://cloud.yippy.com/ - visualizes topics based on search queries. Created at Vivisimo company, also creator of one of the most successful meta search engines, offering classification of search results, Clusty (Vivisimo).

SenseBot Summarizer summarizes search results in the form of a tag cloud.

Google on the other hand offers ”Wonder wheel” option, in order to display search results

TagCloud Summarizer is a tool that users can use to instantly visualize a topic using the familiar tag cloud display. Users can create a cloud based on a query.

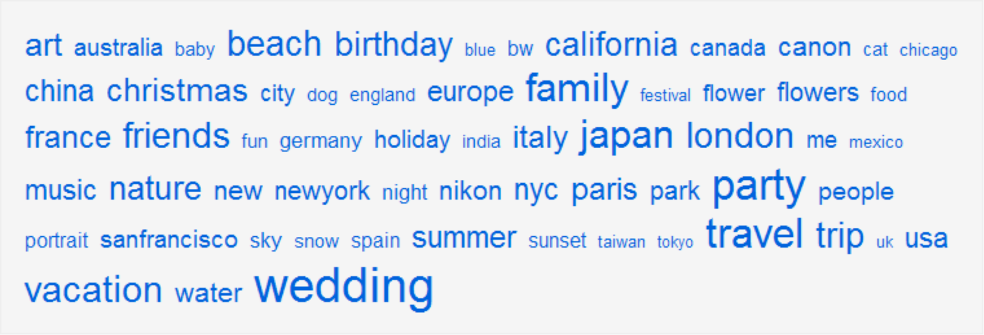
TagCloud Summarizer generates a cloud using the user's search results for the topic they enter. Using the Summarizer to generate the cloud also ensures that it is always up-to-date because topics/main concepts are generated in real-time, based on the user's query.

4.2 use

Use the TagCloud Summarizer for online web-pages, search systems, or personal web-sites.

Summary. *This chapter presents an overview of tag-clouds used as a method for representing text content.*

Tag Clouds are popular applications used for vaious purposes: as a navigation mechanism, as indicators of activity within social media experiences, for visualization in texts and textual data, for annotation of documents 4.1. The importance or weight of words in the tag cloud are shown with size of font and/or color. The tag clouds are hyperlinks leading to a collection of items associated with the tag.
A version of tag cloud is called text cloud. It is used as a visual display that conveys the broad themes that emerge from textual analysis. There are three types of tag clouds depeding on their purpose and use. The first type contains a tag represeting the frequency of each term. The second type is a global tag cloud whose tags has frequencies aggreggated over all items and users. The third type of tag cloud contains categories, and its tags' size indicates the number of subcategories.



art australia baby beach birthday blue bw california canada canon cat chicago china christmas city dog england europe family festival flower flowers food france friends fun germany holiday india italy japan london me mexico music nature new newyork night nikon nyc paris park party people portrait sanfrancisco sky snow spain summer sunset taiwan tokyo travel trip uk usa vacation water wedding

Figure 4.1: Tag Cloud

4.3 1

Related work
Opinion Crawl[1] - web sentiment analysis application. It generates a concept cloud from daily scanned blogs, web site articles.

[1] http://www.opinioncrawl.com/

SenseBot Search Results Summarizer is a plugin for Mozilla Firefox browser that generates a tag cloud of the main concepts returned as search results from Google.

Search Cloudlet[2] is another Firefox Addon that inserts a related tagcloud into Google interface. Working behind the scenes, Search Cloudlet injects a tag cloud of related words in to both Google and Yahoo search results pages. Then you can use the tag links to quickly and easily filter and refine your searches.
LinkSensor SenseBotSummarizer All three are based on SenseBot - a semantic search engine. Made available from Semantic Firefox Extensions [3]

4.4 Brainstorming

What is a tag cloud? Graphical representation of a collection of tags. Tag clouds visualize word frequency in a given text.

Tag clouds may be used as a topic summary.

There are three main types of tag cloud applications used in social software.

1. frequency of items / tags
2. number of items to which a tag has been applied
3. tags are categorization method for content items

The following tag clouds were evaluated in order to select the solution that is most applicable for Tag Cloud Summarizer project.

- TagsTreeMaps[4]
- OpenCloud[5]

state:
- LSA has low precision performance but handles nicely synonymy problem

[2] https://addons.mozilla.org/en-US/firefox/addon/9943/
[3] http://www.semanticengines.com/plugins.htm
[4] http://tagstreemaps.sourceforge.net/TagsTreeMaps.html
[5] http://opencloud.sourceforge.net/

- what is ontology, what are document annotations
- what problems have dms search, lsa
- visualization by Tag Cloud

Challenges

A drawback of the classical LSA implementation as an information retrieval 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, Mr. Mugo constructs a concept-document matrix from the annotations used, and concatenates it with the term-document matrix normally generated in LSA process. The proposed solution, however, results in a slow speed of LSA, and has left Mr. Mugo's hypothesis open.

Objective

The objectives of this work are to investigate how to improve the precision of LSA method by using semantic annotations, and how to adequately present the results of LSA in the form of a tag cloud.

The current project has several objectives. It will investigate the implementation of LSA method for improving information retrieval in a domain-specific DMS 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 relevance feedback techniques.

Semantic annotations are meta data annotations used to add information to unstructured data, in our case to the document collection. Semantic annotations are based on an ontology (what is an ontology and where am i going to describe it?), especially developed for the domain of interest - CoreMedia CMS domain. ...

Relevance feedback is a technique from IR field, used to improve the precision of the method. (what is relevance feedback?)

It will be done by annotating the document collection with semantic annotations, based on an ontology created for the specific domain. The second technique which will be used for influencing LSA precision, is relevance feedback

4.5 Use case

For the purpose of the current project, the investigations introduced in Chapter 1.4 will be performed in the environment of DocMachine 2.0[6], the document management system of CoreMedia AG[7], Hamburg. The system is based on CoreMedia CMS 2008, and it contains manuals and guides which document CoreMedia CMS.

The use case investigated in the current project includes the implementation of a domain-specific IR in a DMS. LSA will be implemented as a IR method to enable concept based search in a document management system having an implementation of full-text search on the document collection.
- enterprise IR system
- LSA implementation
- semantic annotation

Generally, in order for the IR system to be well-designed, the person who is deploying the system must have a good understanding of the document collection, the users, and their likely information needs and usage patterns [1].

What is semantic search?
Semantic search is the application of semantic technologies to information retrieval (IR) tasks [12]. Semantic technologies include expressive ontologies, resource description languages, scalable repositories, reasoning engines and information extraction techniques. The main topics in the field of semantic search are
- achieve expressive description of resources using conceptual representation of the actual resources (e.g. by ontologies), and annotations by semantic web languages (e.g. OWL).
- adapting IR search methods to search in RDF/OWL data, folksonomies. Search is focused on metadata (possibly linked or embedded in textual information).
- complement IR systems containing document collections using semantic technologies.

Semantic search can be divided into three main branches - expressive description of resources, IR technologies for RDF/OWL data, and semantic technologies complement existing IRsystems on document collections.

[6] https://documentation.coremedia.com/
[7] http://www.coremedia.com/

Unsolved tasks/problems in the area of semantic search:

- how to use semantic technologies to capture the information need of the user?
- translate information need of the user to expressive formal queries, user doesn't need to know the difficult query syntax
- extract expressive resource descriptions from documents
- store and query efficiently expressive resource descriptions on a large scale
- handle vague information needs and incomplete resource descriptions
- evaluate semantic search systems and compare them to standard IR systems

Intelligent semantic search - query expansion:
In classical IR query expansion consists of two complementary steps:
- query expansion ;
- terma re-weighting.

Personalized retrieval widens the notion of information need to comprise implicit user needs, not directly conveyed by the user in terms of explicit information requiests. Personalization is an improvement in IR and semantic search. Personalization is a means to improve the performance retrieval (e.g. measured in terms of precision and relevance)as subjectively perceived by users [13].
What is a semantic repository:
A semantic repository is an engine similar to DBMS, even though there is no agreed upon and well-defined term. Semantic repository has the following synonyms: reasoner, ontology server, semantic store, metastore, RDF database. Semantic repositories allow for storage, querying and management of Semantic respositories use ontologies as semantic schemata. Semantic repositories work with flexible and generic physical datamodels (graphs). This gives the opportuntiy to easily interpret and adopt "on the fly" new ontologies or metadata schemata. Sesame is a popular semantic repository that supports RDF(S) and the major query languages related to it. OWLIM is a repository is another repository, that works with Sesame.

Semantic search finds implementation in semantic search engines, such as Hakia[8] or SWSE[9], to name just two of them. Semantic search may be further used for personalization of search results.

[8] www.hakia.com

[9] http://swse.deri.org/

Semantic search promises to provide more precise results than present-day keyword search. While the definition of semantic search may vary, its goal is to provide better search results. Semantic search offers related search results. Semantic search engines attempt to present search results based on context. Semantic search implementations involve many areas, from semantic search engines llike Hakia[10] or SWSE[11], to
- an ontological semantic and natural language processing based search engine.

What is a content management system?

From Sylvia:
- There is a growing demand to find the "right" documents in Internet.
- Many new systems and technologies are developed to enable finding documents through semantic search. Shortly present which articles exist on improving semantic search (Personalization?) Here also explain the notion of semantic search.
- Introduce which systems are implemented for this purpose - e.g. a Document Management System. Describe shortly its structure. (The Document Management System consists of)
- Describe shortly CMS (only enough so that I can start something..to explain!?)

What is semantic search?
- Improve the current Document Management System (DMS) - A semantic structure is necessary so that more "correct" documents are found - David Mugo[quote] showed in his work that with the help of LSA (method for indexing), more documents are found (shortly describe the work of David Mugo). (Present shortly LSA - only so much that the reader will understand later). Give reference to chapter LSA-theorietical description. - Shortly explain other related works in the field.
- The previous works show the following deficiencies (drawbacks,problems,insufficiencies): enumerate the problems.
- Show/demonstrate what is necessary to improve the search. Here define why is the use of Tag Cloud important for the user.

- Solve the problems mentioned earlier.
- Structure of the work.
The objective of this work is to investigate the implementation of semantic search methods into word-based search, and to offer representation of

[10] www.hakia.com
[11] http://swse.deri.org/

the results via a tag cloud.

CoreMedia AG[12] is a company situated in Hamburg, which develops a content management system named CoreMedia CMS. Documentation is an important part of the software development process, and CoreMedia has developed its own editorial system, which provides online user access to CoreMedia CMS guides. The online documentation system, or DocMachine 2.0[13], is based on CoreMedia CMS 2008 and consists of a management or production environment, and delivery or live environment. A simplified overview of DocMachine can be seen in Figure REMOVED.
Using the editor, CMS users can create, edit or delete content, which is managed and stored by the Content Management Server. Content is presented to the end users, who can access the document collection and search through it according to their information need. CMS usually consists of an environment for content management, and an environment for content delivery or presentation. However, at this point we will only introduce the basic concepts of the system. For more detailed introduction, please refer to

DocMachine uses Apache Lucene[14] information retrieval engine for full text indexing, and document retrieval based on full text search. However, Lucene has the limitation that it provides search based on word-matching, thus queries containing ”physician” for example will not return as results documents containing ”doctor”. This limitation of word-matching search technique refers to ”synonymy”, the case when more than one term describes the same concept. To overcome such limitations and provide for a concept-based search, Information Retrieval methods can be used, such as Latent Semantic Analysis(LSA).

Latent Semantic Analysis (LSA) is a theory and method for extracting and representing the contextual-usage meaning of words by statistical computations applied to a large corpus of text.[14] LSA is interesting with respect to document retrieval, as it allows for a search based on meaning. It was developed as an information retrieval technique to improve upon the common procedure of matching words of queries with words of documents. The method exploits statistical properties of term distribution among documents to overcome the common problem of word sense ambiguity. For that, the documents are mapped to vectors in a continuous vector space. Then, the dimensionality of the original data is

[12] https://www.coremedia.com/

[13] https://documentation.coremedia.com/

[14] http://lucene.apache.org/

reduced to uncover the latent semantic structure by using a linear algebra method called Singuar Value Decomposition, or SVD. The retrieval and comparison of the documents are performed on the reduced data.
The classical implementation of LSA as an information retrieval method has certain drawbacks. Generation of the SVD matrix is computationally expensive, and inclusion of new doocuments in the document collection requires re-generation of the term-document matrix, and re-computation of the term and document weights. Another drawback is the low precision of returned results.

A previous work investigating the improvement of LSA precision performance by including document anotations in LSA method, is "Connecting people using latent semantic analysis for knowledgee sharing" by David Mugo.[2] In his work, Mr. Mugo constructs a concept-document matrix from the ontological annotations used in the documents, and concatenates it with the term-document matrix normally generated in LSA process. The proposed solution, however, results in slow speed of LSA, and has left Mr. Mugo's hypothesis open.
The motivation of the current work is to investigate the improvement of DocMachine's search functionality by implementing LSA as a document retrieval technique. However, considering the low precision that LSA displays, it is worth investigating in the direction of improving LSA precision performance. Therefore, the objective of this work is to reseach the improving of LSA precision performance by applying ontology-based semantic annotations to documents, and including these annotations into LSA process. To visualize the results from LSA, a Tag cloud will be constructed from the terms in the reduced term-document matrix. This Tag cloud will enable the user to associate through drag-and-drop terms from the tag cloud with paragraphs from the documents in the document collection, thus applying higher weights to the terms used. After the user input, LSA will perform again on the document collection, and LSA precision performance will be evaluated.
The reminder of this work is organized as follows. Chapter **??** describes in more detail DocMachine, the editorial system used at CoreMedia AG. Chapter 4 presents the basic concepts of ontologies and document annotations based on ontologies. Chapter 2 discusses Latent Semantic Analysis method, its deficiencies, and presents an approach for improving LSA's precision by including semantic annotations in the method. Chapter 5 presents the implementation and makes an evaluation of the results achieved in this work. And finally, conclusions are drawn in Chapter 6, along with some limitations from the current study and outlook for a future research.

Chapter 5

Implementation and evaluation

Summary. *This chapter reports the implemented solution for the given thesis problem, gives discusses its advantages and disadvantages.*

It is a challenge by itself to come up with a sensible evaluation set for an IR implementation. ... Define here precision, recall, measures , how to measure LSA performance with diff. k, clusters, cluster labelling.

The document collection consists of guides and manuals about CoreMedia CMS 5.2.

5.1 LSA implementation

LSA was applied to a collection of 11818 words in 4000 documents, all of which describe CoreMedia CMS 5.2. The algorithm took 63552 ms for preprocessing and indexing of the whole document collection.

For the implementation of LSA this work uses the open LSA library which is part of Semantic Spaces Project[15]. It is developed at the Natural Language Processing Group at the University of California at Berkley (UCLA)[1].

The real difficulty of LSA is to find out how many dimensions to remove - the problem of dimensionality.

[1] `http://code.google.com/p/airhead-research/`

TODO:test if inluding only terms that occur in more than one document improved LSA performance with respect to generating precise tag clouds.

5.2 Tag Cloud implementation

The implemented open source library used for tag cloud generation is called Opencloud[2], and is provided by Marco Cavallo.

5.3 Tools used

Airhead Research[3] project was used as a semantic spaces Package which provides a java-based implementation of LSA.

Apache Lucene[4] was used as an indexing and search library.

5.4 Advantages and drawbacks

why am i using lsa instead of lda for example?

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

5.5 Latest development in the field of LSA

LSAView is a tool for visual exploration of latent semantic modelling, developed at Sandia National Laboratories [16].

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

5.6 Improvements

only terms occuring in more than one documents have been included
stop words

[2] http://opencloud.mcavallo.org/

[3] http://code.google.com/p/airhead-research/

[4] http://lucene.apache.org/java/3_0_2/

compound words list
Lanczos algorithm for computation of SVD of large sparse matrices
Multi-threading computation in the project

Chapter 6

Conclusion and outlook

Summary. *summarize me*

6.1 1

6.2 Future Work

1. implement LSA based IR in full-text search results, or as recommendation.
2. link tags from tag cloud to the documents where they occur (using lucene indexing ???)
3. investigate also the case of updating the document collection - how to handle this case
4. Improve TagCloudSummarizer to work also with German texts (company has a website that support German, Russian, French..)
5. Make the process run in parallel.

Acronyms

CMS Content Management Systems.

IR Information Retrieval.

LSA Latent Semantic Analysis.

SVD Singular Value Decomposition.

Appendix A

Appendix

TODO: insert important source code parts here.

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

[9] 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.

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

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

[12] “Proceedings of the Workshop on semantic Search (SemSearch 2008) at the 5th European Semantic Web Conference (eswc 2008), tenerife, spain, june 2nd, 2008,” in *SemSearch* (S. Bloehdorn, M. Grobelnik, P. Mika, and D. T. Tran, eds.), CEUR Workshop Proceedings, CEUR-WS.org, 2008.

[13] A. Micarelli and F. Sciarrone, “Anatomy and empirical evaluation of an adaptive web-based information filtering system,” *User Modeling and User-Adapted Interaction*, vol. 14, no. 2-3, pp. 159–200, 2004.

[14] T. Landauer, P. Foltz, and D. Laham, “An introduction to latent semantic analysis,” *Discourse processes*, vol. 25, pp. 259–284, 1998.

[15] D. Jurgens and K. Stevens, “The S-Space package: an open source package for word space models,” in *ACL ’10: Proceedings of the ACL 2010 System Demonstrations*, (Morristown, NJ, USA), pp. 30–35, Association for Computational Linguistics, 2010.

[16] 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.