HAMBURG UNIVERSITY OF TECHNOLOGY
INSTITUTE FOR SOFTWARE SYSTEMS

# An Investigation of Latent Semantic Mapping of Ontologies

Diploma thesis by Pawel Kazakow

Hamburg, 27th June 2008

Supervisor: Prof. Dr. Ralf Möller
Second Supervisor: Prof. Dr. Helmut Weberpals
Advisor: M. Sc. Atila Kaya

# Declaration

I hereby confirm that I have authored this thesis independently and without use of others than the indicated resources. All passages taken out of publications or other sources are marked as such.

Hamburg, 27th June 2008

City, Date Sign

# Acknowledgements

I sincerely would like to thank Prof. Dr. Ralf Möller for providing me with an interesting and challenging topic of research and for giving me the opportunity to perform this work in his department. Special thanks to M. Sc. Atila Kaya for his support and motivation. I also thank all my family and friends for their support during the making of this thesis and the whole study.

# Abstract

In this thesis, applications of Latent Semantic Analysis on ontologies have been investigated. Psychological foundations of knowledge modelling has been explored to better understand the relationship between Latent Semantic Analysis and the Semantic Web, and to indicate the limitations of knowledge-based technologies compared to the human mind. The Semantic Web models knowledge in an explicit way using ontologies, networks of interrelated concepts. By contrast, Latent Semantic Analysis models knowledge in an implicit way by mapping documents to a continuous vector space, and reducing the dimensionality of the data. The functional principle of Latent Semantic Analysis and the underlying singular value decomposition have been investigated and visually explained. The generalisation of the approach to Latent Semantic Mapping and the requirements of the data suitable for this analysis have been described. Methods for graph analysis and data mining in relational databases have been proposed, introducing the novel term Latent Semantic Data Mining (LSDM). Based on these methods, approaches for probabilistic reasoning have been derived.

In dieser Arbeit wurden Anwendungen von Latent Semantic Analysis (Latente Semantische Analyse) auf Ontologien untersucht. Psychologische Grundlagen der Wissensmodellierung wurden erkundet, um den Zusammenhang zwischen Latent Semantic Analysis und dem Semantischen Web besser zu verstehen und auf die Grenzen der wissensbasierten Technologien verglichen mit der menschlichen Psyche hinzuweisen. Das Semantische Web nutzt zur expliziten Wissensmodellierung Ontologien, Netzwerke von in Beziehung stehenden Konzepten. Im Gegensatz dazu modelliert Latent Semantic Analysis Wissen auf implizite Weise durch Abbildung von Dokumenten auf Vektoren in einem kontinuierlichen Vektorraum. Das Funktionsprinzip von Latent Semantic Analysis und der zugrunde liegenden Singulärwertzerlegung wurde untersucht und visuell erklärt. Die Verallgemeinerung des Ansatzes auf Latent Semantic Mapping (Latente Semantische Abbildung) und die Anforderungen an die für die Analyse geeigneten Daten wurden beschrieben. Methoden für Graphanalyse and Data Mining in relationalen Datenbanken wurden vorgeschlagen, wobei ein neuer Fachbegriff Latent Semantic Data Mining (LSDM) eingeführt wurde. Basierend auf diesen Methoden wurden Ansätze für probabilistisches Reasoning abgeleitet.

# Contents

# 1. Introduction

The staggering progress in computer technology in the past decades has revolutionised our lives. The computer technology became ubiquitous. The Internet has changed the way we communicate and do business, look for information and entertain ourselves. The increasing storage capacity and processing speed allowed to build more sophisticated applications enabling people to achieve better and more results in less time. Meanwhile, the ever dropping hardware prices, despite of the increasing power, enabled more people to access that technology and to contribute. The volume of information available on the Web is growing at an exponential rate, amplifying the need for intelligent text and language processing. The Semantic Web vision is the evolving worldwide web of data, extending the current Web, enabling computers and humans to better work in cooperation, and helping to manage the complexity and volume of the available information.

## 1.1. Motivation and Objective

The rapid advancement of computers and the Internet make science fiction visions about learning and thinking machines achieving human-level intelligence appear to become more realistic than ever, raising a question about the actual cognitive abilities, machines can potentially reach. In particular, for estimating the potential of the Semantic Web vision, it is important to understand, what computers are theoretically able to learn from the entire information available on the current Web.

My interest was to become acquainted with upcoming knowledge-based technologies, such as the Semantic Web, and to investigate the limitations of those compared to the human mind. The initial idea, proposed by the professor, to investigate Latent Semantic Analysis in the context of the Semantic Web gave me a great starting point to develop an exciting thesis.

The Semantic Web models knowledge in an explicit way using ontologies, networks of interrelated concepts. The vision of the Semantic Web is a worldwide web of data, extending the current Web, enabling computers and humans to better work in cooperation.

Latent Semantic Analysis 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 reduced to uncover the latent semantic structure. The retrieval and comparison of the documents are performed on the reduced data.

The objective of this thesis is to investigate applications of Latent Semantic Analysis on ontologies, later labelled as Latent Semantic Mapping of ontologies. Moreover, psychological foundations of knowledge modelling is to be explored to better understand the relationship between Latent Semantic Analysis and the Semantic Web, and to indicate the limitations of knowledge-based technologies compared to the human mind.

## 1.2. Structure

Psychological foundations of knowledge modelling are covered in Chapter 2. The subsequent Chapter 3 provides an overview of human memory models, suggesting that Latent Semantic Analysis and the Semantic Web can be considered as such.

The functioning principle of Latent Semantic Analysis and the underlying singular value decomposition are investigated and visually explained in Chapter 4.

In Chapter 5, the generalisation of Latent Semantic Analysis to Latent Semantic Mapping and the requirements of the data suitable for this analysis are described. Methods for graph analysis and data mining in relational databases are proposed, resulting in approaches for probabilistic reasoning.

The covered material and the achieved results are discussed in Chapter 6. Chapter 7 summarises the results, and provides ideas for the future work.

# 2. Human Memory

The amount of digitally available information on the Web keeps growing at an astounding pace. However, most Web pages are designed for human consumption, while computers are used only to display the information. Search engines do not interpret the search results, human intervention is still required for that. This situation is progressively getting worse, as the increasing size of the search results produces information overflow. To cope with this problem, computers must understand the information. This understanding requires human like common knowledge about the world. In general, the results produced by natural language technologies, such as machine translation, proofreading, and speech recognition and synthesis, get significantly improved when common knowledge is used.

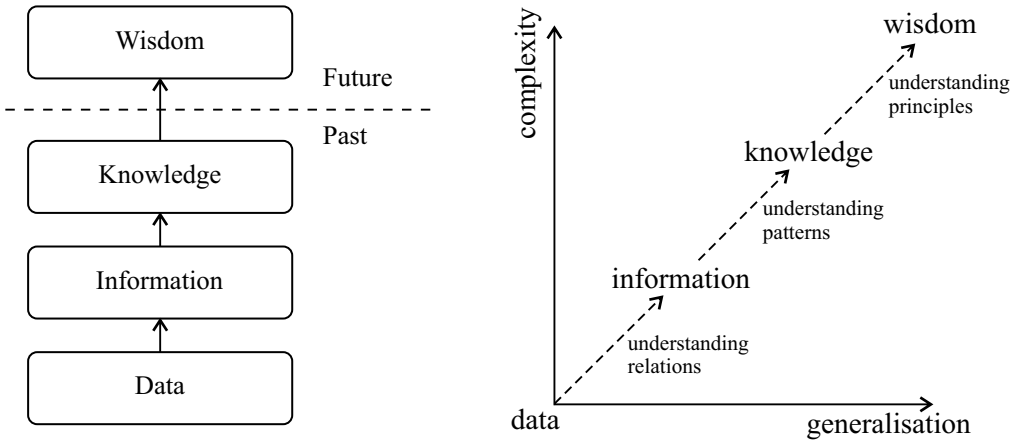
This chapter provides a basis to understand the nature of knowledge a computer can possess and to better estimate the limitations of knowledge-based technologies compared to human mind that are often being unrealistic influenced by the future visions suggested by science fiction.

## 2.1. Knowledge

The ever increasing power of computers has made us to understand the capabilities and limitations of our mind better and to rethink the conventional definitions of Data, Information and Knowledge, defining those terms at a higher level of abstraction.

The content of the human memory can be classified into four categories [4]: *Data*, *Information*, *Knowledge* and *Wisdom* (DIKW). The hierarchy of these categories is illustrated in Figure 2.1(a). The first three categories relate to the past, to what has been or what is known. Only the fourth category, wisdom, relates to the future because it incorporates vision and design. Wisdom gives people the ability to create the future rather than just grasp the present and past.

The diagram illustrated in Figure 2.1(b) gives an alternative perspective, pointing out that the higher the level in the DIKW hierarchy, the higher the complexity and the generalisation. In the following, we give definitions of the categories in the DIKW hierarchy based




(a) The DIKW hierarchy, after [4]

(b) Complexity and generalisation, after [5]

Figure 2.1.: Content of the human memory

on [37], [5], and [4] in particular. The terms are explicated with examples afterwards.

*Data* is raw. It simply exists and has no significance beyond its existence. It can exist in any form, usable or not. It does not have meaning of itself.

*Information* is data that has been given meaning by way of relational connection. This meaning can be useful, but does not have to be. Information provides answers to *who*, *what*, *where*, and *when* questions.

*Knowledge* is a deterministic, interpolative and probabilistic process. It is cognitive and analytical and implies understanding. It is the process by which one can synthesise new knowledge and information from the previously held knowledge. Knowledge is application of data and information and provides answers to *how* questions and an appreciation of *why*. Memorised information does not become knowledge without understanding.

*Wisdom* is an extrapolative and non-deterministic, non-probabilistic process. It beckons to give understanding about which there has previously been no understanding, and in doing so, goes far beyond understanding itself. It is an evaluated understanding. It involves concepts and relations on a very abstract level. It gives the power to make decisions, the ability to design the future by visualising and taking action. Wisdom is a uniquely human state, a machine can never reach.

Data is the result of perception of the world with our senses. At this level of the hierarchy, a text document comprises a collection of meaningless compositions of letters and signs.

A reader who speaks the language of the document can understand the meaning of words and sentences, and thus acquire information. The reader would gain knowledge from it by contemplating on the collected information. The difference between information and knowledge is the difference between memorising and learning.

In the early history computers were data processing machines, predominantly used to process business data. Today we live in the information age where computers store and process data to supply us with information. The amount of the available information is growing at an exponential rate making it virtually impossible for humans to manage its complexity. The demand for technologies with the ability to filter the relevant information is rapidly increasing. This requires machines to build human like knowledge from information involving a certain level of understanding. Evolving technologies like the Semantic Web attempt to address certain aspects of the new requirements.

## 2.2. Memory Structure

Human brain is one of the most intriguing entities known today. Progress in neurophysiology has repeatedly proven that the neurochemistry of the brain is much more complex than previously expected; artificial neural networks provide only a very rough simulation of the actual processes in the human brain, leading to learning abilities and intelligent behaviour. In contrast to neurophysiologists, psychologists attempt to explain the mind and the brain in the context of real life, studying mental processes and behaviour. However, there is a little area of overlap between neurophysiology and psychology at present, such that neurophysiology is of marginal relevance for psychological studies [8]. This section briefly describes the structure of the human memory from the psychological perspective.

A basic and generally accepted classification of the memory is based on the duration of memory retention and identifies three distinct types of memory: sensory memory, short-term memory and long-term memory. This classification is well represented in the model illustrated in Figure 2.2.

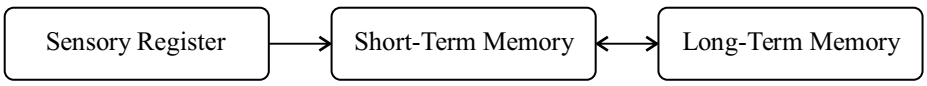



Figure 2.2.: Memory types in the multi-store model, adopted from [7]

There is a *sensory register* for each of the five senses that stores information for several hundred milliseconds, before it goes to the short-term memory. The visual and auditory

registers are broadly studied in cognitive psychology due to simple setup for reproducible experiments. The visual register helps us to see a continuous movement when watching a film, or to perceive continuous figures when painting them in midair in the dark with a torch. The auditory register helps us to recognise repetitions in noise.

From sensory register, information is partly transferred to the *short-term memory*. It allows one to recall this information from several seconds to a minute without rehearsal and is able to hold up to five items, like numbers to dial a phone. The short-term memory is also referred to as *working memory*, since it is also used to process information recalled from the long-term memory [8].

### 2.2.1. Long-Term Memory

The storage in sensory memory and short-term memory have a strictly limited capacity and duration, which means that information is available for a certain period of time, but is not retained indefinitely. In contrast, *long-term memory* can store much larger quantities of information for potentially unlimited duration. Long-term memory has an immensely high complexity as it stores everything we know about the world, our whole life experience and all our skills.

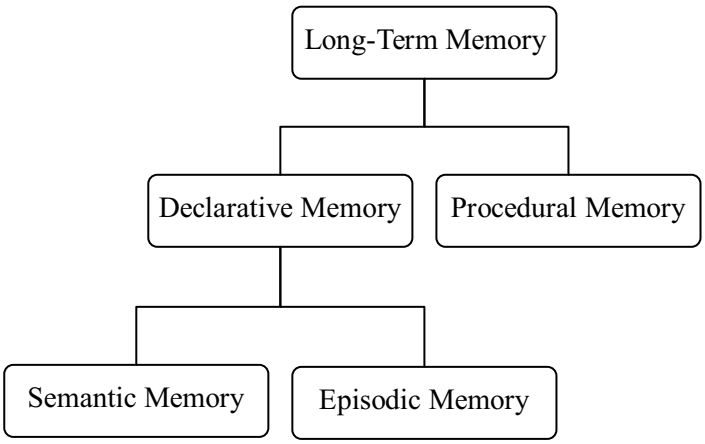



Figure 2.3.: Long-term memory structure

The structure of the long-term memory is illustrated in Figure 2.3. It is divided in declarative (explicit) and procedural (implicit) memories [6].

*Procedural memory* stores skills and procedures, and can be used without consciously thinking about it. It can reflect simple stimulus response pairing or more extensive

patterns learnt over time.

*Declarative memory* stores facts that can be consciously discussed, or declared. It is divided in episodic memory and semantic memory.

*Episodic memory* refers to the memory of events, times, places, associated emotions, and other memory in relation to an experience.

*Semantic memory* refers to the memory of meanings, understandings, and other conceptual knowledge unrelated to specific experiences. The conscious recollection of factual information and general knowledge about the world generally thought to be independent of context and personal relevance. Semantic memory includes generalised knowledge that does not involve memory of a specific event.

Skills like driving an automobile are stored in the procedural memory. A particular event of driving is content of the episodic memory. The fact that an automobile is a wheeled passenger vehicle is stored in the semantic memory. Semantic memory thus refers to general facts and meanings we share with others, whereas episodic memory refers to unique and concrete personal experiences.

A skill stored in the procedural memory cannot be communicated; one cannot learn to drive an automobile without practise. The content of the episodic memory cannot be communicated either; sharing an experience means communicating rather the semantic description of it. Consequently, semantic memory is the only memory that can actually be communicated, and thus can potentially be acquired from a text document, written in natural language, by a computer. Knowledge-based technologies use models of the semantic human memory.

## 2.3. Summary

The exponentially growing amount of information overwhelms humans. To filter the relevant information better, and in general to enhance natural language technologies, computers are required to understand the processed information and, for that, possess human like common knowledge. The DIKW hierarchy helps to better understand the terms data, information, knowledge and wisdom. It makes clear that computers have to operate on the higher level of this hierarchy; information technology has to evolve towards knowledge technology to fulfil the new requirements.

A machine can never acquire full human knowledge without having a human body to ex-

perience life, without being human [32]. The structure of the human memory suggests that semantic memory is the only memory that can be actually communicated. It is thus the only type of human memory a computer can potentially acquire from text documents. Knowledge technologies are based on models of the semantic memory. The semantic knowledge gives a computer an abstract comprehension of concepts expressed by a natural language allowing a certain level of understanding of the processed information. This, in turn, allows a computer to interpret and to filter information based on conceptual relevance.

# 3. Semantic Memory Models

Human semantic memory stores conceptual knowledge about the world unrelated to specific personal experiences that may have led to this knowledge. Semantic memory thus stores knowledge about objects and events, about language, and about how language is used to refer to objects and events. Knowledge-based technologies use computational cognitive semantic memory models that describe the functions of human memory in implementable mathematical detail. Numerous models of the semantic memory can be classified into four types shown in Figure 3.1, each of which accurately, though not completely, models various aspects of empirically observed properties of the human memory [3]. These aspects involve both the structures of knowledge and the processes that operate on these structures.

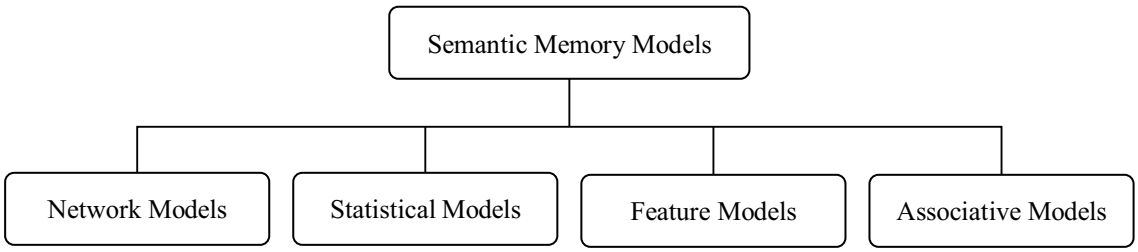



Figure 3.1.: Semantic memory model types

## 3.1. Network Models

Network models organise knowledge in networks, composed of a set of interconnected nodes, representing interrelated concepts and hierarchical relations. The connection links may be weighted to reflect the strength of a relation.

### 3.1.1. Teachable Language Comprehender

In early attempts to use computers to translate from one language to another, programs performed mechanical substitution of words in the first language with those in the second language and arranged the word order so as to conform to the grammar of the second language. The results were often disastrous, and it soon became obvious that successful translation depends on comprehending the thoughts expressed by the sentence. Comprehension, in turn, depends on world knowledge that is not contained in a sentence. Psychologists, also concerned with the process of comprehension, joined artificial intelligence researchers in working on the problem of representing and retrieving world knowledge.

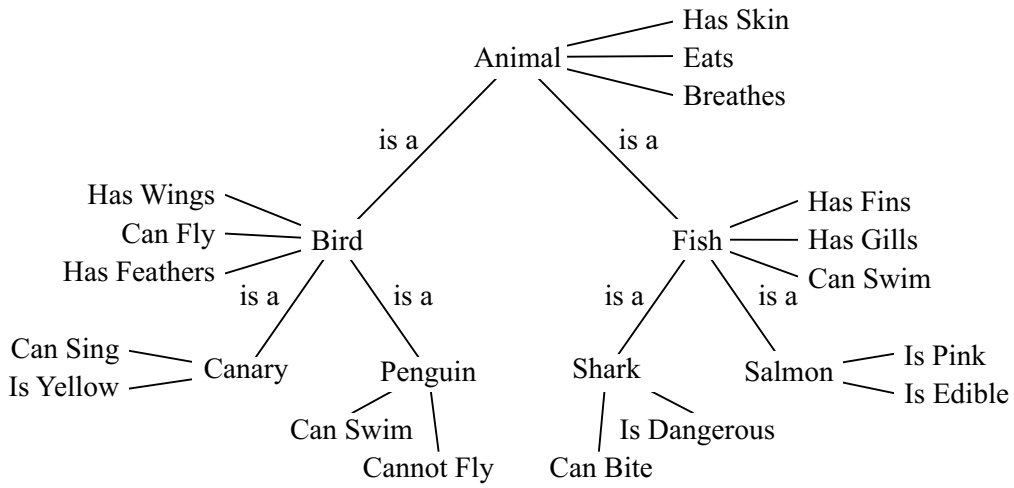



Figure 3.2.: Part of the hierarchical, semantic network, after [8]

Motivated by the challenges of machine translation, the Teachable Language Comprehender (TLC) was one of the first attempts to model human knowledge on a computer with the goal of recreating human inferential ability. The underlying model by Collins and Quillian, demonstrated in Figure 3.2, organises knowledge into a hierarchical network of concepts and attributes connected by relations. Each concept is linked to at least one superordinate concept, inheriting all its attributes. The inheritance of attributes prevents redundancy and is referred to as *cognitive economy* [8, 22].

### 3.1.2. Semantic Web

The World Wide Web today may be defined as the *Syntactic Web* or the web of Documents, where information presentation is carried out by computers, and the interpretation

and identification of relevant information is delegated to humans. This interpretation process is very demanding and requires great effort to evaluate, classify, and filter relevant information. The volume of available digital data is growing at an exponential rate, and it is becoming virtually impossible for humans to manage the complexity and volume of the available information. The Web has evolved as a medium for information exchange among people, rather than machines. Consequently, the semantic content, which is the meaning of the information in a Web page, is coded in a way that is accessible to humans alone.

The *Semantic Web* or the web of data is an evolving extension of the Web, in which information is given well-defined meaning, better enabling computers and humans to work in cooperation. The fundamental idea of the Semantic Web is to deposit the meaning of the statements made in a document from natural language and to formulate explicitly in a machine readable way, and add this information to the document, thus expanding the Web to include more machine-understandable resources. Similar to the Syntactic Web, it should be as decentralised as possible. The promise of the Semantic Web is to unburden human users from cumbersome and time-consuming tasks [15, 12].

### 3.1.3. WordNet

*WordNet* is a large lexical database of English. Nouns, verbs, adjectives and adverbs are grouped into *synsets*, sets of cognitive synonyms, each expressing a distinct concept. Synsets are interlinked by means of conceptual-semantic and lexical relations, resulting in a network of meaningfully related words and concepts. *WordNet* is an important resource freely and publicly available to researchers in computational linguistics, text analysis, and related areas.

## 3.2. Statistical Models

Statistical memory models acquire knowledge as a form of statistical inference from a discrete set of units distributed across a number of compositions resulting in an occurrence matrix. The semantic structure is inferred by applying statistical analysis to that matrix.

## 3.3. Latent Semantic Analysis

Mainly used in information retrieval, Latent Semantic Analysis has also demonstrated the ability to model human semantic memory by learning from a large corpus of representative English text. With the produced knowledge base, Latent Semantic Analysis have achieved promising results in a variety of language tests [28]. It has been reported in [28] that LSA has produced promising results in a variety of language tests, such as a TOEFL[1] vocabulary test.

Further techniques, such as Probabilistic LSA (PLSA) [20] and Latent Dirichlet Allocation (LDA) [14], evolved from Latent Semantic Analysis adding sounder probabilistic models. An investigation of those techniques would significantly expand the scope of the thesis.

## 3.4. Feature Models

In contrast to network-based models, feature-based models elaborate on processes that operate in semantic memory but make minimal assumptions about the structuring of knowledge. In the feature comparison model proposed by Smith et al. [40] concepts are described by relatively unstructured sets of attributes, called *semantic features*. Those features are classified in *defining features*, which are essential to defining the concept, and *characteristic features*, which are often associated with a concept but are not essential to its definition. Defining features are attributes shared by all members of a category. In contrast, characteristic features are attributes shared by many, but not all, members of a category. The number of features decreases as the concept becomes more superordinate.

The similarity of concepts depends on the number of the shared features. A statement in a sentence is verified by comparing the feature sets that represent its subject and predicate concepts. The characteristic feature of this model type is the absence of an explicit structure. The following example demonstrates how the concepts *Bird* and *Penguin* from Figure 3.2 would be represented in a feature-based model:

$$\begin{aligned} \textit{Bird} &: \{\textit{Has Skin}, \textit{Eats}, \textit{Breathes}, \textit{Has Wings}, \textit{Can Fly}, \textit{Has Feathers}\} \\ \textit{Penguin} &: \{\textit{Has Skin}, \textit{Eats}, \textit{Breathes}, \textit{Has Wings}, \textit{Cannot Fly}, \textit{Has Feathers}, \textit{Can Swim}\} \end{aligned}$$

The implicit hierarchy can be indirectly computed by feature comparison, and it has been

[1] Test of English as a Foreign Language

demonstrated that feature models can be translated directly into network models and vice versa [21]. More recent theories have accepted that categories may have a fuzzy structure [30], rather than distinct membership determined by logical rules for the combination of features.

## 3.5. Associative Models

Associative memory models describe the semantic memory as a set of concepts and the strength of their association. The corresponding mathematical structure is the association graph with weighted edges, which can be described by a quadratic adjacency matrix. Each element of the matrix corresponds to the strength of the association between the corresponding concepts. In contrast to network models, the association graph does not explicitly describe any hierarchy. The topology of a semantic network can be represented by an acyclic graph.

Search of Associative Memory (SAM) [36] and neural networks are examples of associative memory models.

## 3.6. Summary

The computational models described in this chapter successfully mimic specific characteristics of human memory, though, they do not generalise well and fail outside the boundaries of their assumed conditions. With the goal of performing a full range of human cognitive tasks, a more general approach has been the development of *cognitive systems*, comprised of cognitively justified tools and theoretical constraints; they are used to develop and test new cognitive models.

Although more complete in their description of human cognition, cognitive systems are too broad and powerful to serve as the basis of a knowledge-based information retrieval system. Together with associative models, cognitive architectures are beyond the scope of information management; for the storage and retrieval of information objects thus more computationally tractable models have been chosen [22].

All semantic memory models overlap in their basis; they all describe relationships between information pieces, a fundamental concept in psychology. Each model thus can be represented by a graph or a matrix, allowing a variety of mathematical methods to be applied

for analysis. Latent Semantic Analysis uses the singular value decomposition to reduce the dimensionality of the matrix.

# 4. Latent Semantic Analysis

*Latent Semantic Analysis* (LSA) is a theory and method for analysing global relationships between textual data objects and the terms they contain. In the specific context of information retrieval, LSA is often referred to as *Latent Semantic Indexing* (LSI) [16].

In information retrieval, two widely used measures are *recall* (4.1) and *precision* (4.2). Recall shows the ability of a retrieval system to present all relevant items, while precision shows its ability to present only relevant items.

$$\text{recall} := \frac{\text{number of relevant documents retrieved}}{\text{total number or relevant documents}} \tag{4.1}$$

$$\text{precision} := \frac{\text{number of relevant documents retrieved}}{\text{total number or retrieved documents}} \tag{4.2}$$

The problem is that users typically retrieve documents on the basis of *conceptual* content, and individual words provide unreliable evidence about the conceptual topic or meaning of a document due to word sense ambiguity, a pervasive characteristic of natural languages. There are usually many ways to express a given concept (synonymy). The literal terms in a query may not match those of a relevant document, resulting in poor recall performance. In addition, many words have multiple meanings (polysemy). The literal terms in a query may match terms in documents that are not of interest to the user, decreasing precision performance. Stated more formally, the information needs of people are in the concept space, while keyword based access to information operates in the word space. Words represent concepts but the mapping from words to concepts is ambiguous. This problem is known as *word sense disambiguation* (WSD). In terms of the DIKW hierarchy (see 2.1), words and syntax correspond rather to the data and information level, while concepts and semantics correspond rather to the knowledge level of the hierarchy.

LSA was developed as an attempt to improve on the common procedure of matching words of queries with words of documents. For that, the information is treated in the *statistical* domain by taking advantage of higher order implicit *semantic structure* in the association of terms with documents. The assumption is that this structure is partially obscured by the

randomness of word choice with respect to retrieval. To uncover the latent semantic structure and get rid of the obscuring noise, the dimensionality of the original data is reduced by means of the singular value decomposition. Document retrieval and comparison are then performed on the reduced data.

## 4.1. Vector Space Model

*Vector Space Model* (VSM) is an algebraic model for representing text documents as vectors. The input data is a document set with $n$ documents and a list of $m$ unique terms indexed from those documents. Each document is mapped to a vector $c_j$, $1 \leq j \leq n$, forming the columns of the *occurrence matrix* $W$:

$$W := (c_1, c_2, ..., c_n) = (r_1, r_2, ..., r_m)^T \in \mathbb{R}^{m \times n} \tag{4.3}$$

where $r_i$, $1 \leq i \leq m$ are the rows of the matrix corresponding to the unique terms, and $^T$ denotes transposition.

### 4.1.1. Feature Extraction

The *term frequency* $f_{i,j}$ is the number of times an $i$-th term appears in the a $j$-th document. The elements of the occurrence matrix $w_{i,j}$ incorporate a function of the term frequency $f_{i,j}$.

It is often desirable to normalise the term frequency with a per document factor $\lambda_j$ to prevent a bias towards large documents, which may have a higher term frequency regardless of the actual importance of that term in the document:

$$w_{i,j} = \frac{f_{i,j}}{\lambda_j} \tag{4.4}$$

Depending on the application, different normalisation methods are used:

$$\lambda_j = \|f_j\|_1 := \sum_i |f_{i,j}| \tag{4.5}$$

$$\lambda_j = \|f_j\| := \sqrt{\sum_i f_{i,j}^2} \tag{4.6}$$

where $\|\cdot\|_1$ and $\|\cdot\|$ are the one-norm and the two-norm, respectively.

In the classic vector space model [38] the *term frequency – inverse document frequency* (TF-IDF) is used, which additionally comprises a global parameter to reflect the importance of a term depending on its usage among all documents in the set:

$$w_{i,j} = \log\left(\frac{n}{\sum_j \operatorname{sign}(f_{i,j})}\right) \cdot \frac{f_{i,j}}{\lambda_j} \tag{4.7}$$

$$\operatorname{sign}(x) := \begin{cases} 1, & x > 0 \\ 0, & x = 0 \\ -1, & x < 0 \end{cases} \tag{4.8}$$

where $\sum_j \operatorname{sign}(f_{i,j}) \geq 1$ determines the number of documents the $i$-th term appears in, assuming that the term appears in at least one document. High values for $w_{i,j}$ are reached by a high term frequency within a $j$-th document and a low document frequency among all documents. This model hence filters out common terms (and stopwords), but upweights rare terms to reflect their relative importance.

In case of strong fluctuations and outliers among term frequencies the normalisation causes loss of valuable statistical information due to erasement of low values. The logarithm handles this problem by flattening the local outliers:

$$w_{i,j} = \log\left(\frac{n}{\sum_j \operatorname{sign}(f_{i,j})}\right) \cdot \frac{\log(f_{i,j})}{\lambda_j} \tag{4.9}$$

The use of unit *entropy* $\varepsilon_i$ instead of the inverse document frequency is proposed in a later work that should reflect the local relative importance of terms more accurately [9]:

$$w_{i,j} = (1 - \varepsilon_i)\frac{f_{i,j}}{\lambda_j} \tag{4.10}$$

$$\varepsilon_i = -\frac{1}{\log(n)} \sum_{j=1}^{n} \frac{f_{i,j}}{\tau_i} \log\left(\frac{f_{i,j}}{\tau_i}\right) \tag{4.11}$$

where $\tau_i = \sum_j f_{i,j} \geq 1$ denotes the total number of times the $i$-th term appears among all documents.

### 4.1.2. Similarity Metrics

In the vector space model *angle cosine* is used as the similarity metric:

$$\angle(a,b) := \cos(a,b) = \frac{a^T b}{\|a\| \cdot \|b\|} \tag{4.12}$$

where $a \in \mathbb{R}^m$ and $b \in \mathbb{R}^m$ are two arbitrary vectors, such as document vectors, and $\|\cdot\|$ denotes the two-norm.

The general *distance matrix* $\angle(A,B) \in \mathbb{R}^{r \times s}$ comprises the cosine distances between all column vectors of the matrices $A = (a_1, a_2, ..., a_r) \in \mathbb{R}^{m \times r}$ and $B = (b_1, b_2, ..., b_s) \in \mathbb{R}^{m \times s}$:

$$\begin{aligned} \angle(A,B) :=& (A \cdot \eta_A)^T (B \cdot \eta_B) \\ \eta_A :=& \text{diag}(a_1^T a_1, a_2^T a_2, ..., a_r^T a_r)^{-\frac{1}{2}} \\ \eta_B :=& \text{diag}(b_1^T b_1, b_2^T b_2, ..., b_r^T b_s)^{-\frac{1}{2}} \end{aligned} \tag{4.13}$$

where $\text{diag}(\cdots)$ denotes the diagonal matrix. Hence, the distance matrices $\angle(W,W)$ and $\angle(W^T,W^T)$ comprise the cosine distances between all documents and all terms, respectively:

$$\begin{aligned} \angle(W,W) =& (W \cdot \eta_W)^T (W \cdot \eta_W) = T^T T \\ \eta_W =& \text{diag}\left(c_1^T c_1, c_2^T c_2, ..., c_n^T c_n\right)^{-\frac{1}{2}} \end{aligned} \tag{4.14}$$

$$\begin{aligned} \angle(W^T, W^T) =& (W^T \cdot \eta_{W^T})^T (W^T \cdot \eta_{W^T}) \\ \eta_{W^T} =& \text{diag}\left(r_1^T r_1, r_2^T r_2, ..., r_m^T r_m\right)^{-\frac{1}{2}} \end{aligned} \tag{4.15}$$

where $T$ is the term document matrix. The *term document matrix* is a special distance matrix, which comprises the cosine distances between terms and documents. It is simply the occurrence matrix with the unified column vectors:

$$T := \angle(I, W) = W \cdot \text{diag}\left(c_1^T c_1, c_2^T c_2, ..., c_n^T c_n\right)^{-\frac{1}{2}} \tag{4.16}$$

where $I := \text{diag}(1, 1, ..., 1)$ is the identity matrix.

Alternatively, *correlation* can be used as similarity metric, which indicates the strength of a linear relationship between vectors:

$$\times(a, b) := \frac{(a - \bar{a})^T (b - \bar{b})}{\|a - \bar{a}\| \cdot \|b - \bar{b}\|} \tag{4.17}$$

where $\bar{a} = \frac{1}{m} \sum_i^m a_i$ and $\bar{b} = \frac{1}{m} \sum_i^m b_i$ are the empirical mean values.

### 4.1.3. Document Retrieval

A query is mapped to the vector $q$ the same way as a document. According to the chosen metric, the similarity to each document vector is computed. As a result, the documents are listed in the descending order of their similarity to the query document until a specified threshold is reached.

For the angle cosine the result vector $r \in \mathbb{R}^n$ is the product of the query vector with the term document matrix:

$$r := \angle(q, W) = \frac{q^T}{\|q\|} T \tag{4.18}$$

### 4.1.4. Document Preprocessing

*Stopwords* are language specific common words (e. g. prepositions, pronouns, articles) that do not carry useful information. Several stopword lists for English and other languages are widely available on the Web. Prior to the computation of the document vectors, stopwords are filtered out to reduce the size of the occurrence matrix and thus the computational complexity.

*Stemming* is a process of reducing inflected and derived words to their stem or root form, based on language-specific rules. Without stemming, different inflections of the same term would be processed as distinct terms, resulting in a larger occurrence matrix that reflects the statistical document structure worse. Stemming is performed after the filtering of the stopwords.

```
Stemming process reducing inflected
derived words stem, based
language-specific rules.  Without
stemming, different inflections term
processed distinct terms, resulting
larger occurrence matrix reflects
statistical document structure worse.
Stemming performed filtering stopwords.
```

```
stem process reduc inflect deriv word
stem base languag specif rule without
stem differ inflect term process
distinct term result larger occurr
matrix reflect statist document structur
wors stem perform filter stopword
```

Figure 4.1.: Document preprocessing example

Figure 4.1 demonstrates the results of the document preprocessing. The box on the left contains the antecedent paragraph with the stopwords removed. The box on the right contains the final result after stemming have been applied that used as an input to compute the occurrence matrix.

## 4.2. Singular Value Decomposition

*Singular value decomposition* (SVD) is a technique, closely related to eigenvector decomposition and factor analysis. Suppose $A \in \mathbb{R}^{m \times n}$, then there exists a factorisation of the form:

$$A = U\Sigma V^T = \sum_{i=1}^{n} u_i \sigma_i v_i^T \tag{4.19}$$

where $U = (u_1, u_2, ..., u_m) \in \mathbb{R}^{m \times m}$ and $V = (v_1, v_2, ..., v_n) \in \mathbb{R}^{n \times n}$ are orthogonal matrices; $\Sigma = \text{diag}(\sigma_1, \sigma_2, ..., \sigma_n) \in \mathbb{R}^{m \times n}$ is a diagonal matrix with the *singular values* $\sigma_1 \geq \sigma_2 \geq ... \geq \sigma_n$ in its diagonal, uniquely determined by $A$. This factorisation is denoted as the singular value decomposition of $A$.

### 4.2.1. Dimension Reduction

The deficiency of the vector space model is that it does not handle synonymy and polysemy. LSA introduces dimension reduction by means of the SVD to uncover the latent semantic structure and thereby to partially handle those issues.

The general approach based on reducing the dimensionality of the data is known as *Principal component analysis* (PCA), probably the oldest and best known of the techniques of multivariate analysis. The idea of PCA is to reduce the dimensionality of a multidimensional data set consisting of a large number of statistically dependent variables, while retaining as much as possible of the variation present in the data set. PCA is defined as an orthogonal linear transformation that transforms the data to a new set of variables, the principal components which are uncorrelated, and which are ordered so that the subspace with the greatest variance comes to lie on the first principal component, the subspace with the second greatest variance on the second one and so on, such that the first few principal components retain most of the variation present in all of the original variables. SVD provides a computationally efficient method of actually finding those principal components [24].

The matrix $\Sigma$ in (4.19) is uniquely determined by $A$ and contains the singular values in its diagonal ordered in decreasing fashion. Thereby the first singular value corresponds to the subspace with the greatest variance, the second value to the subspace with the second greatest variance and so on.

The rank of the matrix $\text{rank}(A)$ is determined by the number of linear independent column vectors and is equal to the number of nonzero singular values. The *rank reduction* is performed by nullifying the lowest singular values and reconstructing the matrix with the remaining $k$ highest singular values:

$$A_k = U_k \Sigma_k V_k^T = \sum_{i=1}^{k} u_i \sigma_i v_i^T \tag{4.20}$$

The Frobenius norm $\|A\|_F$ is defined in terms of those values:

$$\|A\|_F = \|U\Sigma V^T\|_F = \|\Sigma V^T\|_F = \|\Sigma\|_F = \sqrt{\sum_{i=1}^{\text{rank}(A)} \sigma_i^2} \tag{4.21}$$

According to the Eckart and Young theorem, $A_k$ is the best rank $k$ approximation of $A$:

$$\|A - A_k\|_F = \min_{\text{rank}(X) \leq k} \|A - X\|_F \tag{4.22}$$

Without loss of generality, suppose $A \in \mathbb{R}^{m \times m}$ a quadratic matrix, $A_k \in \mathbb{R}^{m \times m}$ the best rank $k$ approximation of $A$ and $P \in \mathbb{R}^{m \times m}$ a permutation matrix, then the following statements are true:

$$\tilde{A} := PA \Rightarrow A_k = P^T \tilde{A}_k \tag{4.23}$$

$$\tilde{A} := AP^T \Rightarrow A_k = \tilde{A}_k P \tag{4.24}$$

$$\tilde{A} := A^T \Rightarrow A_k = \tilde{A}_k^T \tag{4.25}$$

meaning that permutation of rows (4.23) or columns (4.24), or the transposition (4.25) does not affect the approximation result.

### 4.2.2. Rank Estimation

The assumption of LSA is that the latent semantic structure within a document collection has a lower dimension than the original occurrence matrix. Rank estimation refers to the choice of the number of retained dimensions, denoted by $k$. Choosing $k$ too low may cause loss of relevant information, while choosing $k$ too high may cause the result to contain too much noise. There is no general procedure known for choosing $k$; it is rather an empirical issue [28] and depends on methods used for the evaluation of the retrieval results.

A possible approach is to analyse the approximation error for all $k$ and to choose the value right before a considerable ascent of it [13]. The distribution of singular values provides an initial estimation of $k$ sometimes.

As an alternative approach we propose to choose $k$ depending on the span between the average inter- and intra-topic distances (see 4.3.2 for more details). However, it is unclear, how those span is related to the retrieval performance.

## 4.3. Topic Decomposition

The objective of *topic decomposition* is to cluster documents depending on their topics. *Document grouping* can be used to retrieve similar documents to the chosen or retrieved one, supporting a research process. Similarly, the objective of *term clustering* is to group terms with similar meaning or context, which can be used to refine a query and thereby resolve polysemy [13].

Topic decomposition and the related data clustering are own fields of research [33], and are discussed here with relevance to LSA and underlying methods.

In the following, two approaches for topic decomposition are presented: the first one is an own approach based on angle cosine, the second one is a data clustering algorithm used for topic decomposition.

### 4.3.1. Angle Threshold

In this thesis, we propose the exploitation of the angle cosine (4.12) in combination with a threshold for topic decomposition. Experiments carried out during this work that selecting $45^\circ$ as a threshold provides reasonable results for topic decomposition:

$$\rho(\angle(a,b)) = \begin{cases} 1, & \angle(a,b) \leq \frac{1}{4}\pi, \\ 0, & \text{else.} \end{cases} \tag{4.26}$$

where $a$ and $b$ are two arbitrary document vectors.

The threshold applied to the angle matrix yields a binary matrix with groups of equal vectors, which are interpreted as topics. The number of topics seems to always be equal

or lower $k$. Threshold does not guarantee disjoint topics, meaning that topics may share documents. Nevertheless, this approach is a simple and very fast method.

### 4.3.2. Single Linkage Clustering

*Single linkage* is a hierarchical data clustering algorithm. The algorithm does not rely on the LSA, but operates on any array of distances. It successively combines cluster pairs with the minimum distance, starting with single element clusters in the first iteration. The distance between clusters is the minimum distance between their elements. A drawback of this algorithm is its complexity of $O(n^2)$, where $n$ represents the length of the distance array [39].

Topic decomposition performed with single linkage clustering provides an enlightening insight into the functional principle of LSA and the underlying PCA/SVD. Using the single linkage clustering algorithm, it is always possible to separate a document set in $k$ disjoint groups, which can be interpreted as topics. The dimension reduction performed in LSA does not appreciably affect the average distance between inter-topic pairs, but dramatically reduces the average distance between intra-topic pairs. LSA thus improves topic separability [10] and increases the similarity between related documents (see 4.5.4 for exemplification).

## 4.4. Visualisation Methods

Several visualisation methods we found particularly suitable for the data involved in LSA. In this section, these methods are briefly presented. Examples for each method can be seen in Figure 4.9. In the subsequent section they will be exploited to visualise experiment results.

### 4.4.1. Greyscale Image

For the visualisation of matrices greyscale images are used, where each pixel corresponds to a cell in a matrix. This visualisation method is very helpful for getting a quick overview of value distribution within a large matrix.

The element values are mapped to 256 greyscale values. Lowest values of the matrix are mapped to black, highest values to white. The higher the value, the lighter the corresponding pixel appears in the image. The colour bar next to the image displays the colour map.

### 4.4.2. Distance Graph

*Distance graph* is a visualisation method suitable for the distance matrices. The distance graph provides a better overview over the clusters of vectors with relatively small distance. Those are not easily recognised in a greyscale image of a large matrix, when the permutation of the vectors is inappropriate.

Each node of the graph corresponds to a vector. The length of an edge connecting two nodes depends on the distance between the two corresponding vectors. The smaller the distance between the vectors is, the closer the corresponding nodes appear in the graph. For the visualisation, a minimum distance threshold is specified to prevent nodes from overlapping. Between the nodes of vectors with a distance above a maximum distance threshold there is no edge. Those nodes are positioned at a certain distance from each other to accentuate the absence of connection.

### 4.4.3. Dendrogram

*Dendrogram* is used to visualise the results of the single linkage clustering algorithm. It is a hierarchical binary cluster tree, which consists of rectangular shaped lines connecting objects in the tree. The width of each shape represents the distance between the two objects being combined.

## 4.5. Experiments

For the experiments in this work, we have implemented LSA in *MATLAB*. Distance graphs have been plotted with *GraphViz* from the output generated in *MATLAB*. All other visualisations are generated directly with *MATLAB*.

### 4.5.1. Term Co-Occurrence

The term co-occurrence stands for the (frequent) appearance of the related terms in a document. This experiment demonstrates, how the dimension reduction uncovers the latent semantic structure conveyed by the term co-occurrence.

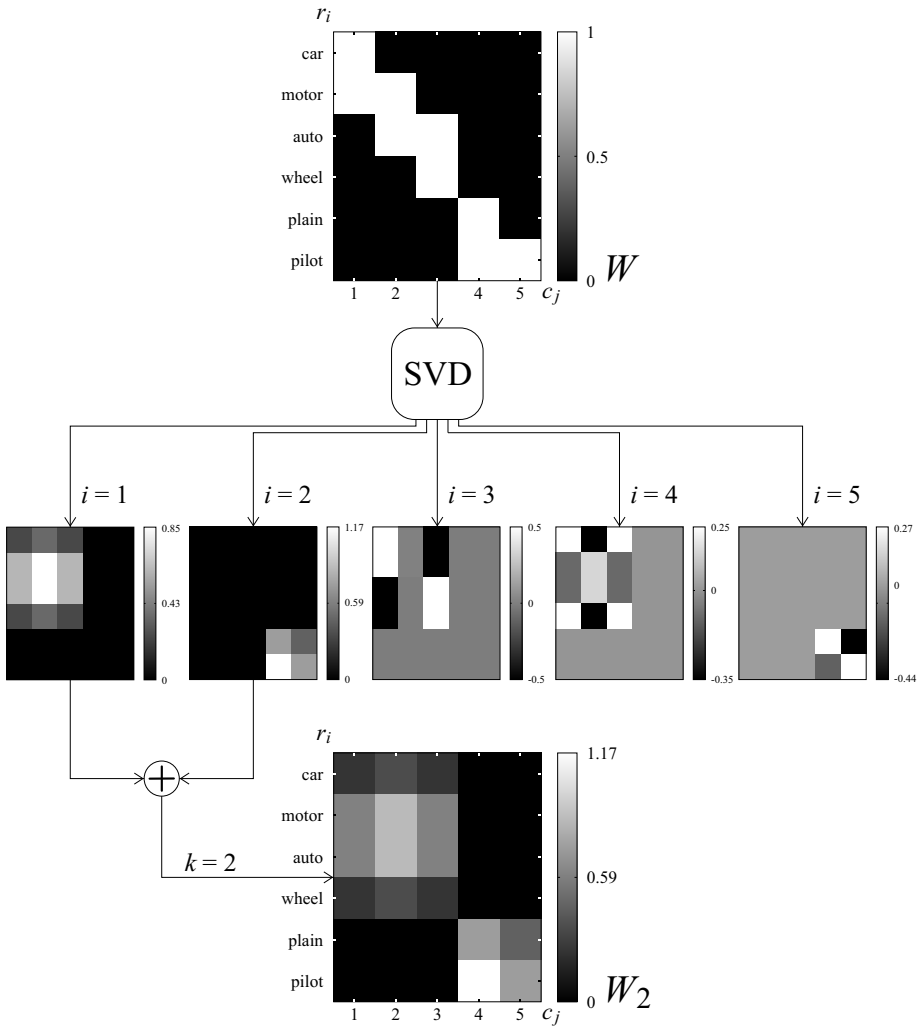



Figure 4.2.: Singular value decomposition and term co-occurrence

In Figure 4.2, the occurrence matrix $W$ of a fictional collection of five documents is visualised. Such a matrix can arise when very short documents like titles are analysed or when taking only the sign (4.8) of the term frequency into account.

For the term *car* the query vector is $q = (1,0,0,0,0,0)^T$. From (4.18) follows the result vector $\text{sign}(r) = (1,0,0,0,0)$. Note that all document vectors in $W$ except for the first one

are orthogonal to the query vector. This means that the first document, which literally contains the term *car*, is the only relevant one for this query.

In this example, the term *motor* co-occurs with *car* in the first and with *auto* in the second document. This implies that the terms *car* and *auto*, which are actually synonyms in this context, have something in common. This fact is not explicitly reflected by the original occurrence matrix $W$.

The singular value decomposition factorises $W$ into matrices $u_i \sigma_i v_i^T$, according to (4.19), which are visualised in Figure 4.2 for all values of $i$. The low rank approximation of $W$ for $k = 2$ is the sum of the first two components:

$$W_2 = \sum_{i=1}^{2} u_i \sigma_i v_i^T = u_1 \sigma_1 v_1^T + u_2 \sigma_2 v_2^T$$

The remaining subspaces are considered as not relevant and are therefore neglected. The resulting matrix $W_2$ exposes the latent semantics in the document collection conveyed by the term co-occurrence. The same query vector $q$ for the term *car* multiplied with the new matrix $W_2$ yields a different result vector $\text{sign}(r) = (1,1,1,0,0)$, which means that the first three documents are relevant for this query. In particular, the second document became relevant, although the search term *car* does not appear in this document but only its synonym *auto*. This means that LSA allows to compute similarity of documents that do not share even a single term.

### 4.5.2. Document Retrieval

The sample document collection taken from [16] consists of the nine technical memoranda listed in Table 4.1. Terms occurring in more than one title were selected for indexing and are *emphasised*. This document collection can be decomposed in two topics: documents $\{1,2,3,4,5\}$ are about human-computer interaction, and $\{6,7,8,9\}$ are about graph theory.

#### Occurrence Matrix

The occurrence matrix $W$ shown in Table 4.2 comprises non-normalised term frequencies.

Table 4.1.: A document collection

| $j$ | Document Content |
|---|---|
| 1 | *human* machine *interface* for abc *computer* applications |
| 2 | a *survey* of *user* opinion of computer system response time |
| 3 | the *eps user interface* management *system* |
| 4 | *system* and *human system* engineering testing of *eps* |
| 5 | relation of *user* perceived *response time* to error measurement |
| 6 | the generation of random, binary, ordered *trees* |
| 7 | the intersection *graph* of paths in *trees* |
| 8 | *graph minors* iv: widths of *trees* and well-quasi-ordering |
| 9 | *graph minors*: a *survey* |

Table 4.2.: The occurrence matrix $W$ of the document collection in Table 4.1

| $i$ | Terms | $c_1$ | $c_2$ | $c_3$ | $c_4$ | $c_5$ | $c_6$ | $c_7$ | $c_8$ | $c_9$ |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | human | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| 2 | interface | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 3 | computer | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 4 | user | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 |
| 5 | system | 0 | 1 | 1 | 2 | 0 | 0 | 0 | 0 | 0 |
| 6 | response | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| 7 | time | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| 8 | eps | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| 9 | survey | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 10 | trees | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 |
| 11 | graph | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 |
| 12 | minors | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |

Table 4.3.: Search results for the term *human*

| $i$ | $r\uparrow$ | Document Content |
|---|---|---|
| 2 | 0.19 | a survey of user opinion of computer system response time |
| 3 | 0.18 | the eps user interface management system |
| 4 | 0.15 | system and human system engineering testing of eps |
| 5 | 0.14 | relation of user perceived response time to error measurement |
| 1 | 0.10 | human machine interface for abc computer applications |
| 9 | 0.02 | graph minors: a survey |
| 8 | -0.01 | graph minors iv: widths of trees and well-quasi-ordering |
| 7 | -0.01 | the intersection graph of paths in trees |
| 6 | -0.01 | the generation of random, binary, ordered trees |

The two clusters can already be recognised in $\tilde{W}$ visualised in Figure 4.3 before the rank reduction: the block of size $9 \times 5$ on the top left identifies the first cluster, while the block of size $4 \times 4$ on the bottom right corresponds to the second one. Depending on the application, the clusters consist of column or row vectors. Apart from permutation, that block structure is typical for an occurrence matrix.

#### Distance Matrices

The distance matrix $\angle(\tilde{W})$ visualised in Figure 4.3 largely reveals the two topics of the document collection: it has a block structure with one block of size $5 \times 5$ on the top and one of size $4 \times 4$ on the bottom, although the blocks are noisy. Correlation matrices often look similar to angle matrices. In this case the correlation matrix $\times(\tilde{W})$ contains more noise than the matrix of angles, but the aforementioned block structure is still recognisable.

In addition, Figure 4.3 illustrates the effect of the term frequency normalisation according to (4.4) with (4.6). The low rank approximation of the occurrence matrix with raw term frequencies $\tilde{W}_2$ exhibit more noise than the approximation of the occurrence matrix with normalised frequencies $W_2$.

#### Document Retrieval

The query vector for *human* is $q = (1,0,...,0)^T$. Table 4.3 lists the search results for *human* in descending order, computed according to (4.18). Despite the fact that the first document in the list does not contain *human* but its synonym *user*, this document appears

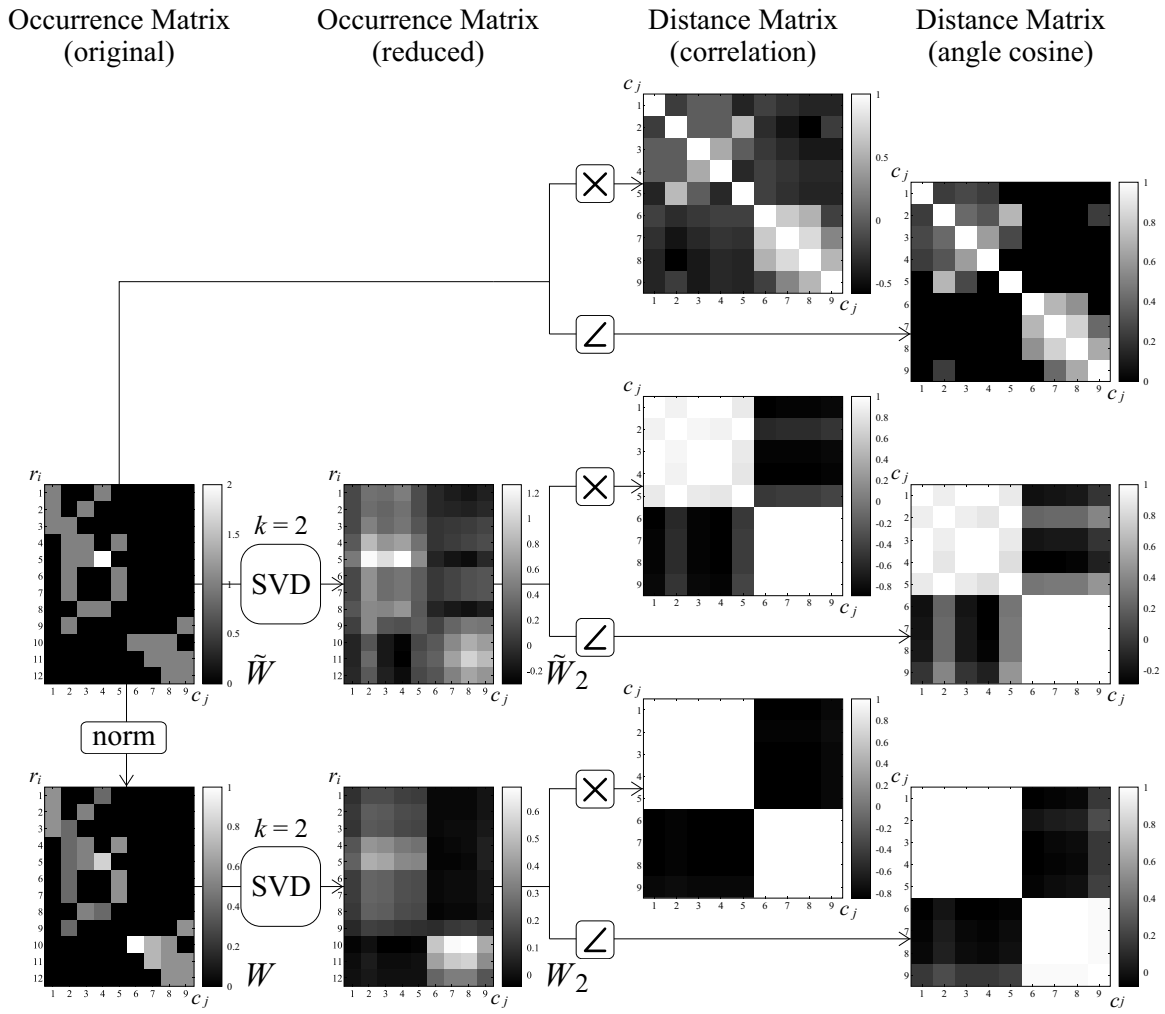



Figure 4.3.: Occurence and distance matrices

first. The last four documents have a noticeably lower similarity value relative to the first five, indicating their irrelevance for the search query.

### 4.5.3. Rank Estimation

The objective of this experiment is to demonstrate rank reduction effects along with the problematic choice of $k$, the number of retained dimensions.

The experiment is performed on typical occurrence matrices, starting with the block matrix $A$ described below, then adding different types of noise to create more realistic situation.

For each matrix the low rank approximation with different values of $k$ is investigated. The experiment is best followed by looking at Figure 4.6.

#### Input Matrix A

As an input data serves the binary $m \times n$ matrix $A$ with $m = n = 32$. For the sake of clarity, the rows and columns of the matrix are ordered in such a way that it exhibits a diagonal block structure.

The blocks $B_1, B_2, B_3, B_4$, being of size $12 \times 6$; $5 \times 11$; $5 \times 5$ and $10 \times 10$, introduce different amount of variance to the data. The four topics corresponding to the blocks are disjoint, meaning that documents of distinct topics do not share terms. Since $\text{rank}(A) = 4$, there are only four nonzero singular values and it follows:

$$A = U\Sigma V^T = \sum_{i=1}^{m} u_i \sigma_i v_i^T = \sum_{i=1}^{4} u_i \sigma_i v_i^T = U_4 \Sigma_4 V_4^T = A_4$$

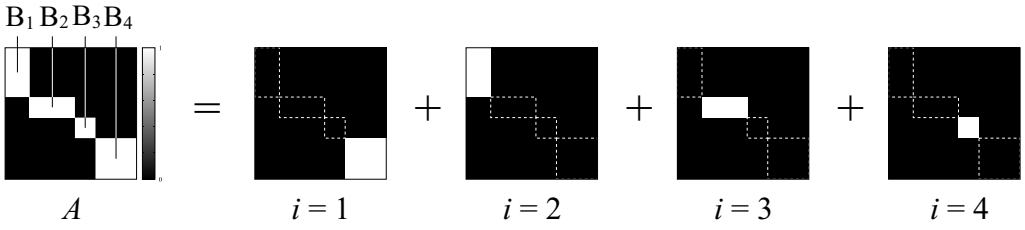



Figure 4.4.: Input matrix A and its singular value decomposition

Figure 4.4 visualises the input matrix $A$ and its singular value decomposition. The greatest singular value $\sigma_1$ corresponds to the subspace of $B_4$, the elements corresponding to this block thus introduce the most variance to the data. The second greatest singular value $\sigma_2$ corresponds to the subspace of $B_1$, this block introduces the second greatest variance to the data, and so on:

$$\sigma_1 \to B_4, \quad \sigma_2 \to B_1, \quad \sigma_3 \to B_2, \quad \sigma_4 \to B_3$$

The choice of an appropriate $k$ for the dimension reduction is trivial, regardless of the previous knowledge that there are four blocks present. The approximation with $k = 4$ yields the original matrix $A$. For $k = 3$ the block $B_3$ with the smallest variance introduction

disappears in the approximation $A_3$. For $k = 2$ block $\mathrm{B}_2$ with the next smallest variance introduction disappears in $A_2$ (see Figure 4.6).

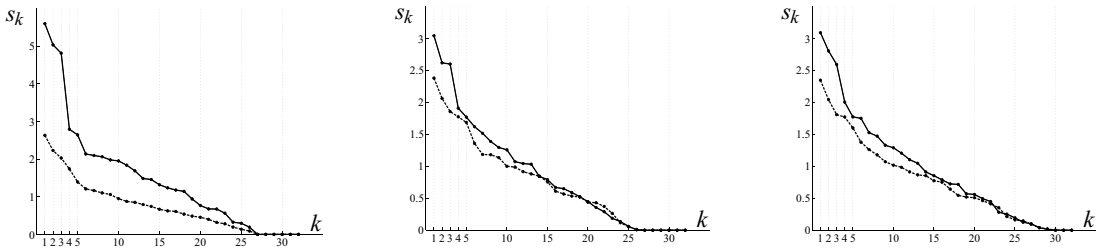
#### Input Matrix B

The next input matrix $B$ is created by binary conjunction of the matrix $A$ with uniformly distributed binary noise $N_\mathrm{B}$:

$$B = \mathrm{and}(A, N_\mathrm{B})$$

There is no noise outside the areas of the blocks in $B$, but the blocks themselves became noisy (see Figure 4.6). The nature of $B$ is very similar to the occurrence matrix from the first example with small documents like titles or sentences, or when taking the sign of the term frequency.

In this case, the choice of an appropriate $k$ without previous knowledge is not obvious anymore. Figure 4.5(a) shows the distribution of the singular values of $B$. Still, there is a noticeable change of slope in the interval $k \in \{3,6\}$. The lower curve shows the singular values for the modified matrix with normalised column vectors. A noticeable change of slope can be observed here for $k = 6$. These observations may be used for an initial estimation of $k$.




(a) Singular values of $B$ (b) Singular values of $C$ (c) Singular values of $D$

Figure 4.5.: Singular values; lower curve: normalised column vectors

The optimum result in terms of the block identification, however, is achieved for $k = 4$ as expected. The four blocks appear in the reconstructed matrix $B_4$, and are visible vividly as solid areas in the correlation matrix $\times(B_4)$.

For $k = 5$ the four blocks are mapped to five subspaces. While in $B_5$ this is hard to detect, a peculiar pattern appears in $\times(B_5)$ in the area corresponding to $\mathrm{B}_4$. This indicates that

the elements of the subspace corresponding to the fifth singular value are scattered in the area of $B_4$ and superimposed with it. Concerning topic decomposition, this means that $B_4$ is separated in two topics.

Block $B_3$ corresponds to the fourth singular value due to its smallest variance introduction. For $k = 3$ all singular values including the fourth one are eliminated, hence block $B_3$ disappears in $B_3$ as expected (see Figure 4.6).

#### Input Matrix C

Binary matrices were analysed so far. Taking the minimum of $B$ and uniformly distributed noise $N_C$ in the interval $[0,1]$, the input matrix $C$ arises:

$$C = \min(B, N_C) = \min(\text{and}(A, N_B), N_C)$$

The blocks in $C$ have the same shape as in $B$ but the values within the blocks are not binary but floating point numbers between zero and one. There is still no noise outside the areas of the blocks, such that the topics remain disjoint (see Figure 4.6).

For $k = 3$ block $B_3$ disappears in $C_3$ as expected but is still not present for $k = 4$ in $C_4$. Instead, pattern present in $\times(C_4)$ indicates that the fourth singular value corresponds to the subspace with the elements scattered in the area of $B_2$ and superimposed with this block. Depending on noise distribution in most cases this does not happen. In this particular case the added noise has decreased the variance introduction of $B_3$ enough to be mapped to a lower subspace than expected. In fact, $B_3$ emerges for $k = 5$ in $C_4$, which indicates that it has been mapped to the subspace corresponding to the fifth instead of the fourth singular value (see Figure 4.6).

Figure 4.5(b) shows the singular values of $C$. There is a considerable drop-off for $k = 3$ to $k = 4$, which may indicate that $k = 4$ is a candidate for the block identification. This is not the case though. In the interval $k \in \{4,6\}$ there is virtually no change of slope. The lower curve shows the singular values for the matrix $C$ with normalised column vectors. Noticeable change of the slope happens for $k = 5$ and $k = 7$ here, demonstrating that the analysis of singular values provide no steady approximation for the choice of $k$. The same is true for the approximation error according to (4.22), since its computation relies on the singular values.

#### Input Matrix D

Finally, the effects of noise outside the former block areas are to be investigated. For that, uniformly distributed noise has been truncated and amplified such that with probability $p = 0.02$ peaks with random amplitude from the interval $[0,1]$ are stored in the matrix $N_{\mathrm{D}}$. Taking the maximum from $C$ and $N_{\mathrm{D}}$ the input matrix $D$ arises (see Figure 4.6):

$$D = \max(C, N_{\mathrm{D}}) = \max(\min(\mathrm{and}(A, N_{\mathrm{B}}), N_{\mathrm{C}}), N_{\mathrm{D}})$$

Since the noise values are taken from the same interval as the values for the blocks, the signal-to-noise ratio of $D$ is low, meaning that the topics are not disjoint anymore but share terms. Consequently, the results of the dimension reduction are disturbed strongly, and the identification of the former blocks and the corresponding topics fails largely (see Figure 4.6).

Similar to the previous matrices, the distribution of the singular values of $D$ shown in Figure 4.5(c) does not provide any firm basis for the estimation of $k$.

### 4.5.4. Topic Decomposition

The objective of this experiment is to demonstrate the methods for topic decomposition, proposed in 4.3. For this experiment the occurrence matrix in Table 4.2 is used.

#### Angle Threshold

Applying angle threshold (4.26) to distance matrices of the approximation with different values of $k$ successively, a topic tree can be built. Distance matrices (angle cosine) and threshold matrices are visualised in Figure 4.9 in the first two columns.

The resulting topic tree is illustrated in Figure 4.7. For the original term document matrix the documents are distributed over seven topics; the documents $\{2,5\}$ and $\{7,8\}$ are grouped. For $k = 8$ a new group with documents $\{6,7\}$ arises that share the document 7. Nothing changes for $k = 7$, for $k = 6$ documents $\{8,9\}$ are grouped and so on. Eventually, for $k = 2$ there are two groups with documents $\{1,2,3,4,5\}$ and $\{6,7,8,9\}$, as expected.

### Single Linkage Clustering

Single linkage clustering is applied to the distance matrices (angle cosine) resulting in a dendrogram for each value of $k$. The plots of the dendrograms are shown in Figure 4.9 in the right column. The objective of this example is to demonstrate that the PCA/SVD separates the documents in $k$ groups and reduces the intra-group distances, while keeping the inter-group distances high. To support the observation, distance graph is plotted next to each dendrogram in Figure 4.9.

The dendrograms show the hierarchical tree of the documents. The document numbers are listed on the vertical axis, and ordered such that the connection lines do not intersect. The width of each rectangular shape represents the distance between the connected clusters. It can be read off from the horizontal axis. The $k$ groups can be read off from a dendrogram by removing $k-1$ links with the greatest width.

For $k=2$ there is one connection line with relatively high width, connecting two groups. The intra-group distances are close to zero, such that the elements of these groups cannot be read off from the dendrogram. According to the distance graph, the elements of the groups are $\{1,2,3,4,5\}$ and $\{6,7,8,9\}$.

For $k=3$ the distance between elements of the first group significantly increases, dividing it in two groups with elements $\{1,3,4\}$ and $\{2,5\}$ with relatively small intra-group distances. The elements of the groups can be easily read both from the dendrogram and from the distance graph.

For $k=4$ there are four groups of documents with relatively high average inter-group distance. The elements of the groups can be read off from the dendrogram and are $\{1,3,4\}$, $\{2,5\}$, $\{6,7,8\}$, and $\{9\}$. In this and other cases for higher values of $k$, the distance graph is not suitable to easily determine the elements of the groups, but still provides an overall impression of the distribution of the distances between elements (see Figure 4.9).

With decreasing value of $k$, the span between average intra-group and inter-group distances increases. The diagram in Figure 4.8 shows the cluster distances for each value of $k$. For $k=2$ there is one cluster with relatively high distance, meaning that there are two document groups present with relatively high inter-group distance. For $k=3$ there are two clusters with relatively high distance. For $k=4$ there are three clusters with relatively high distance, although the span between distances of those three clusters and the remaining five is much smaller.

### 4.5.5. Summary

As an information retrieval technique, LSA improves on the procedure of matching words of queries with words of documents. By resolving synonymy and polysemy it overcomes the problem of word sense ambiguity. Additionally, it has been successfully used in cross-language information retrieval and essay grading. Furthermore, LSA has been used to model human semantic memory by analysing large corpus of representative English text [28].

Documents are mapped to vectors forming the occurrence matrix. Documents sharing many terms are close to each other in the vector space. The latent semantic structure is determined by global correlation patterns. To uncover the latent semantic structure, the rank of the occurrence matrix is reduced. Document retrieval and comparison are then performed on the reduced matrix.

To visualise the functioning principle of LSA, three techniques have been proposed in this thesis: greyscale image, dendrogram and distance graph. The methods themselves are not new. However, no previous work on LSA have been found using those methods in conjunction with LSA. The visualisation has helped to better understand how the matrix is factorised and what the effects of the dimension reduction are.

Important effect of the dimension reduction is the increased topic separability and the increased similarity between related documents. With decreasing number of dimensions retained, the average intra-topic distance decreases while the average inter-topic distance remains largely unaffected.

The choice of the number of retained dimensions is an empirical issue. In this chapter, we have proposed an approach to choose this number depending on the span between the average inter- and intra-topic distances. However, this is a subject of further research (see 7.2.1).

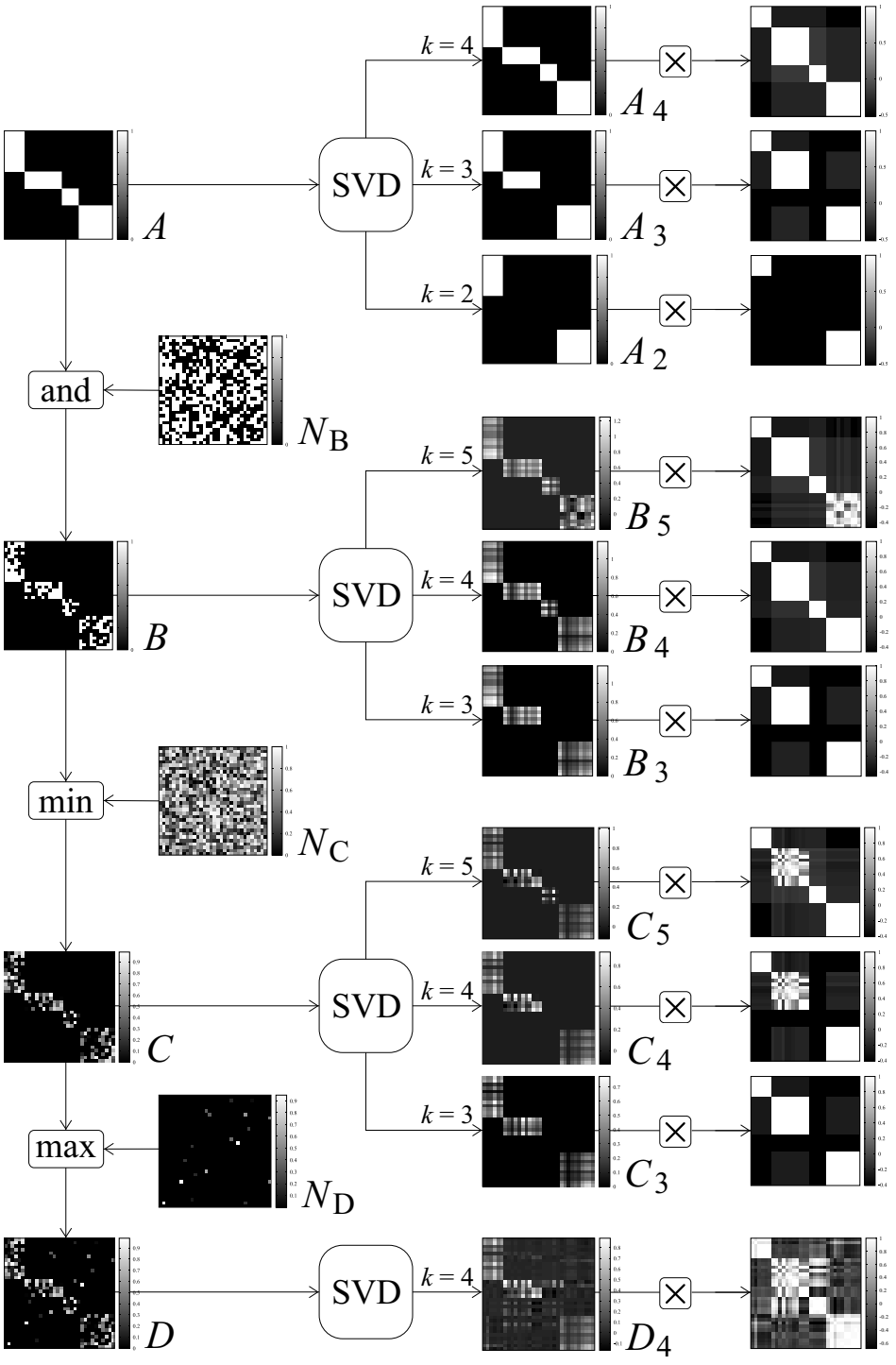



Figure 4.6.: Rank estimation (see 4.5.3 for description)

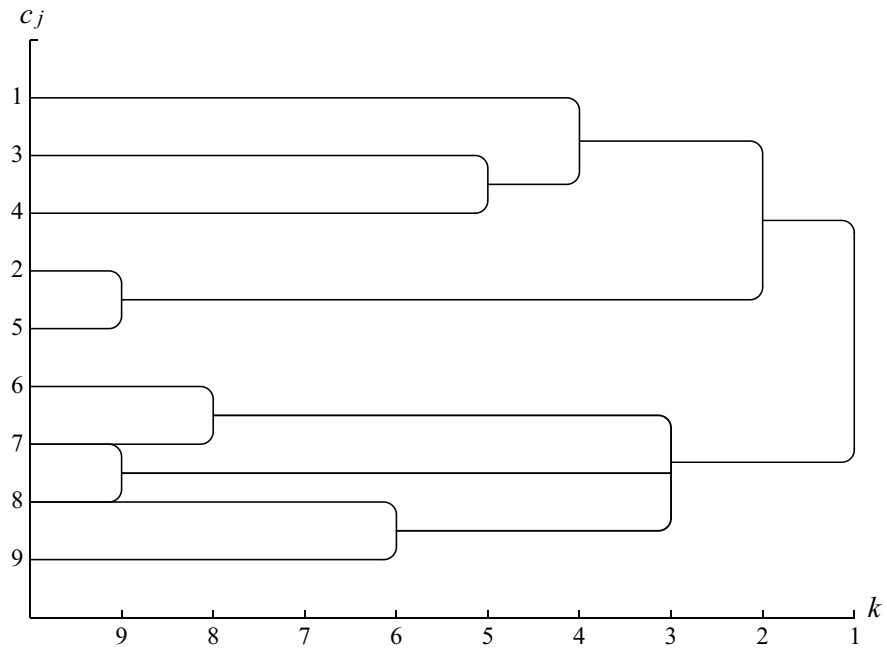



Figure 4.7.: Angle threshold topic decomposition (see 4.5.4 for description)

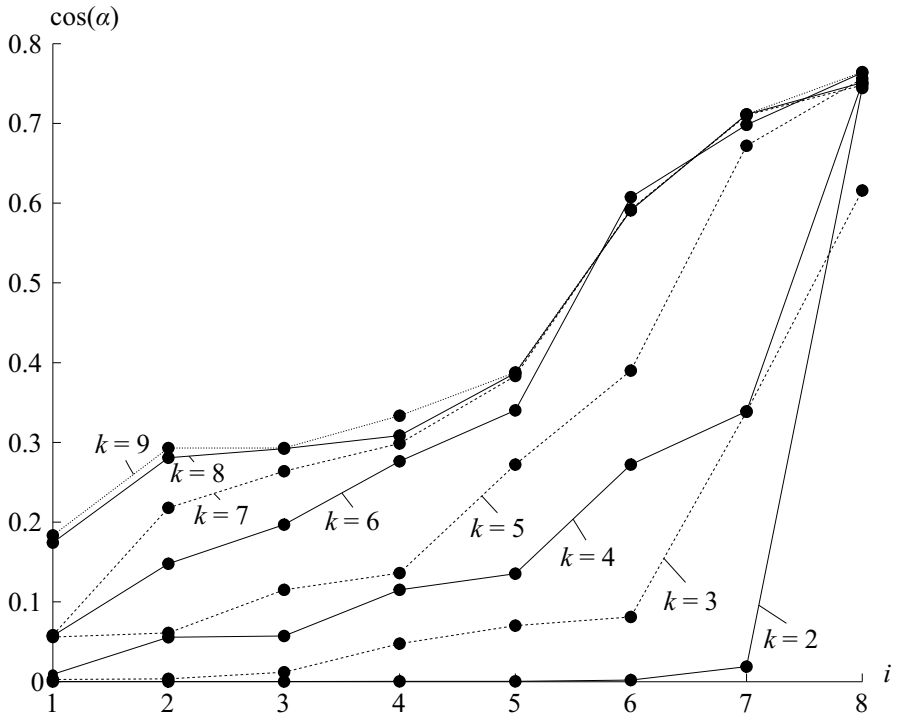



Figure 4.8.: Single linkage cluster distances (see 4.5.4 for description)

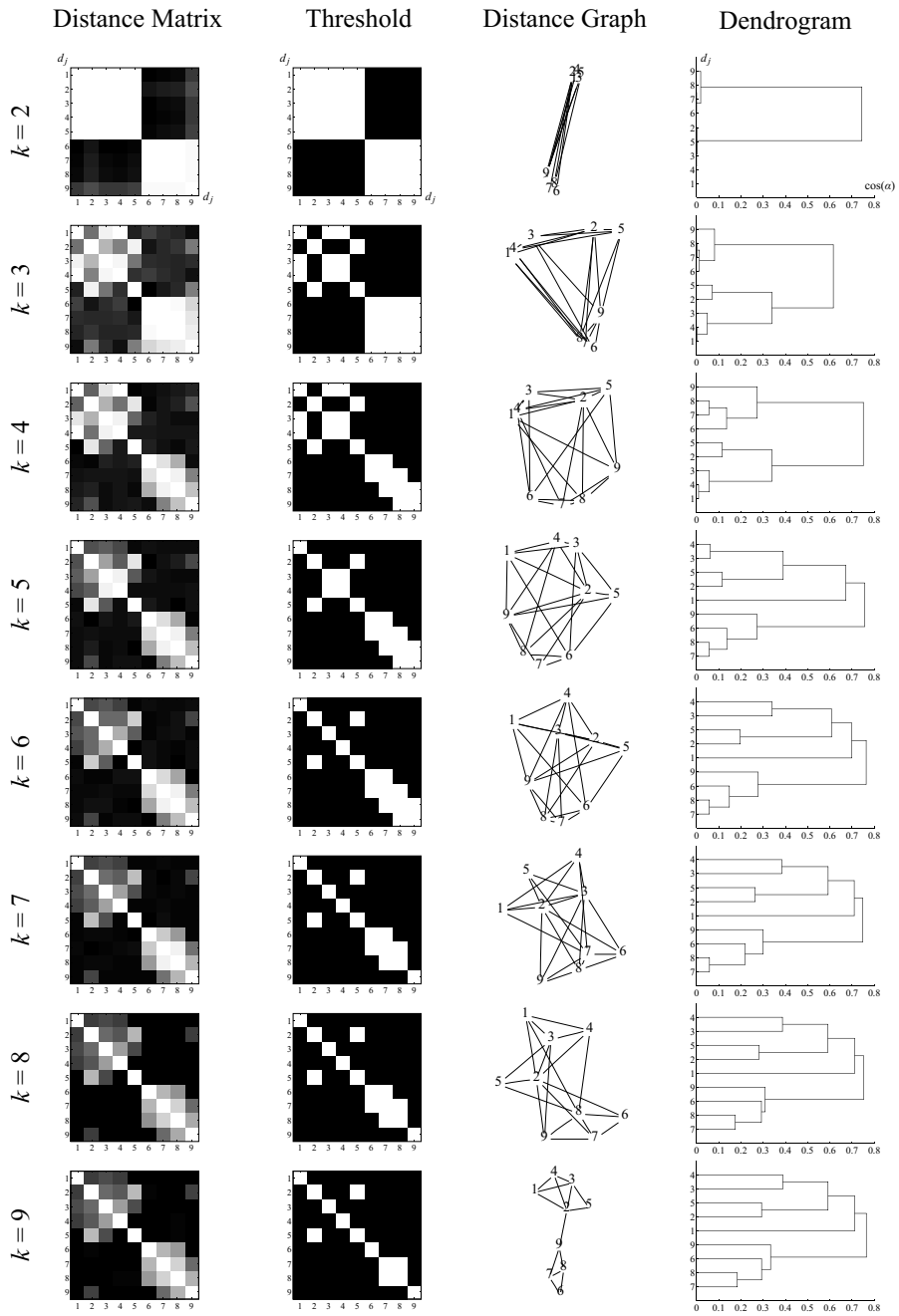



Figure 4.9.: Topic decomposition (see 4.5.4 for description)

# 5. Latent Semantic Mapping of Ontologies

Over the past few years the success of Latent Semantic Analysis in information retrieval led to the application of the same paradigm in many other areas of natural language processing, including large vocabulary speech recognition language modelling and automated call routing [11]. In a distinct application area, bioinformatics, Latent Semantic Structure Indexing (LaSSI) [23] was derived from LSA for calculating chemical similarity and has been recently patented.

In this chapter, the generalisation of Latent Semantic Analysis to Latent Semantic Mapping and the requirements of the data suitable for this analysis are described. Methods for graph analysis and data mining in relational databases are proposed. Based on those methods, approaches for probabilistic reasoning are derived.

## 5.1. Latent Semantic Mapping

Although LSA does not take the word order and sentence grammar into consideration, it has shown the ability to expose global relationships in the language in order to extract useful data concerning topic context and meaning. Three specific factors seem to make LSA particularly attractive [10]: the mapping of discrete entities (in this case, terms and documents) to a continuous parameter space; the dimensionality reduction inherent in the process, which makes complex natural language problems tractable; and the intrinsically global outlook of the approach, which tends to complement the local optimisation performed by more conventional techniques. These are generic properties, which are desirable in a variety of different contexts that are not directly language related. This motivates a change of terminology to *Latent Semantic Mapping* (LSM) [10] to convey increased reliance on the general properties listed above, as opposed to a narrower interpretation in terms of specific topic context and meaning.

LSM generalises a paradigm originally developed to capture hidden word patterns in a set of text documents. Let $\mathbb{M}$ be an inventory of $m$ individual units, such as terms, and $\mathbb{N}$ be a collection of $n$ meaningful compositions of those units, such as documents in a document

Table 5.1.: Examples for Latent Semantic Mapping

| Application (LSM Label) | Units | Compositions |
|---|---|---|
| Information Retrieval (LSA/LSI) [16] | terms | documents |
| Text Summarisation [17] | terms | sentences |
| Junk Email Filtering [11] | words, symbols | emails |
| Speech Recognition [11] | letter n-tuples | words |
| Speech Synthesis [11] | pitch periods | time slices |
| Chemical Similarity Searches (LaSSI) [23] | descriptors | molecules |
| Collaborative Filtering (5.6.3) | unique values | items |
| Ontology Merging (5.6.4) | close terms | class name senses |
| Data Mining (LSDM) (5.4) | unique values | objects |
| Probabilistic Concept Learning (5.5.3) | related concepts | concepts |
| Probabilistic TBox Classification (5.5.3) | semantic features | complex concepts |

set or text corpus. The LSM paradigm defines a mapping between the high-dimensional *discrete* sets $\mathbb{M}$, $\mathbb{N}$ and a *continuous* lower dimensional vector space $\mathbb{L}$ [10].

While mostly conceptually clear, the decision which entities are viewed as units and which as compositions is arbitrary to a certain degree. In LSA for instance, terms can be also viewed as compositions of (occurrence in) documents. In practise, the number of compositions outweigh those of units, such that $n > m$ or even $n \gg m$. The condition $k < \min(m,n)$ in (4.20) must be satisfied in each case.

The underlying PCA/SVD have a much broader field of applications due to its generic properties. The distinctive feature of LSM is the explicit assumption of the underlying latent *semantic* structure in the analysed data, implying the interpretation of the results on the knowledge level of the DIKW hierarchy (see 2.1). Additionally, the feature extraction function is considered as part of the mapping defined by LSM. In contrast, in signal processing SVD is often used to suppress high-frequency noise thus as a low-pass filter. Examples of LSM are illustrated in Table 5.1; methods for LSM of ontologies we propose in this thesis are listed in the last three rows of the table.

## 5.2. Multiple-Type Latent Semantic Mapping

LSM has been successfully used to identify semantic relations between *two* types of objects, entitled as units and compositions of those units. In practical applications however, there are many cases where *multiple* types of objects exist and any pair of these objects could have a pairwise co-occurrence relation. All these co-occurrence relations can be

exploited to alleviate data sparseness or to represent objects more meaningfully.

Multiple-Type Latent Semantic Mapping (M-LSM)[1] is a more recent approach [42]. It is an algorithm which conducts LSM by incorporating all pairwise co-occurrences among multiple types of objects. M-LSA identifies the most salient concepts among the co-occurrence data and represents all the objects in a unified semantic space. In this thesis, M-LSM is briefly described but is not further investigated.

M-LSM analyses the co-occurrence among $q$ types of objects $\{X_1, X_2, ..., X_q\}$, whereas each pair of them could have a pairwise co-occurrence relation. Formally, an undirected graph $G = (V, E)$ is constructed. $V$ consists of $q$ vertices corresponding to each object type. If there is a pairwise co-occurrence relation between two objects, an edge $e_{i,j}$ in $E$ connects the corresponding nodes. Each object type $X_i$ corresponds to a set of $|X_i|$ objects of this type. Each edge thus corresponds to an occurrence matrix $W_{i,j}$ of size $|X_i| \times |X_j|$. Each edge could have a weight $\alpha_{i,j}$ to measure the importance of the relation between $X_i$ and $X_j$. The absence of an edge means that corresponding co-occurrence data is unavailable or not meaningful for an application.

All matrices $W_{i,j}$ and weights $\alpha_{i,j}$ are combined in the unified occurrence matrix $R$, similar to the adjacency matrix of a graph:

$$R = \begin{pmatrix} 0 & \alpha_{1,2}W_{1,2} & \cdots & \alpha_{1,q}W_{1,q} \\ \alpha_{2,1}W_{2,1} & 0 & \cdots & \alpha_{2,q}W_{2,q} \\ \vdots & \vdots & \ddots & \vdots \\ \alpha_{q,1}W_{q,1} & \alpha_{q,2}W_{q,2} & \cdots & 0 \end{pmatrix} \tag{5.1}$$

where $W_{i,j} = W_{j,i}^T$ and $\alpha_{i,j} = \alpha_{j,i}$.

LSM is a special case of M-LSM for $q = 2$ with only one occurrence matrix $W$. In this case the unified occurrence matrix in M-LSM would be:

$$R = \begin{pmatrix} 0 & W \\ W^T & 0 \end{pmatrix}$$

It has been shown in [42] that M-LSM outperforms LSM on multiple applications, including collaborative filtering, text clustering, and text categorisation [42].

[1] The algorithm is originally labelled as Multiple-Type Latent Semantic Analysis (M-LSA). For consistency of terminology, M-LSM is used in this thesis, as it is a generalisation of LSM.

## 5.3. Graph Analysis

Associative models describe the semantic memory as an association graph (see 3.5), which is a mathematical structure generally used to model relations between objects from a finite collection. The topology of a semantic network can be also represented by a graph.

A *graph* $G = (V, E)$ consists of a vertex set $V$ and an edge set $E$. An edge connects two vertices. A finite graph can be described by the *adjacency matrix*. For a directed or undirected graph with $n$ vertices this is a $n \times n$ matrix $E$. The element $e_{i,j}$ takes the value one if there is an edge from vertex $v_i$ to vertex $v_j$, otherwise the element is zero. The adjacency matrix of an undirected graph is symmetric.

### 5.3.1. Graph Partitioning

*Graph partitioning* problem in graph theory consists of dividing $V$ into $k$ parts $V_1, V_2, ..., V_k$ such that the parts are disjoint, and the number of edges with endpoints in different parts is minimised. From the LSM paradigm it is easy to see that rank reduction of the adjacency matrix is well suitable for the graph partitioning problem.

Graph partitioning can be used to separate a circuit amongst circuit boards connected by as few wires as possible. In an ontology, this approach can be used for partitioning of large ontologies, which is an important task in developing scalable reasoning services.

### 5.3.2. Node Clustering

A slightly different application is *node clustering*. From a graph, two subsets of nodes $V_c$ and $V_u$ are chosen, such that these subsets are disjoint. Nodes from the subset $V_c$ are to be clustered. The clustering of nodes depends on the similarities in the sets of edges connecting nodes from $V_c$ to nodes in $V_u$. The nodes in the subset $V_c$ can be thus considered as compositions of nodes in the subset of units $V_u$ (see 5.6.1 for exemplification).

## 5.4. Latent Semantic Data Mining

*Data mining* is a process of finding new, interesting, implicit, potentially useful patterns from very large volumes of data. Data mining is often set in the broader context of *knowledge discovery in databases* (KDD). The KDD process involves selecting and preprocessing the target data, and performing data mining to extract patterns and unsuspected relationships and then interpreting and assessing the discovered structures [34, 18].

From this perspective, LSA can be considered as a method for text mining, while LSM is well suitable for statistical data mining, to discover global correlation patterns within a relational database. For the application of LSM on databases, we introduce the novel term *Latent Semantic Data Mining* (LSDM) to convey the relation of LSM to data mining.

A *database* is a structured collection of datasets. A *dataset* is a collection of objects of the same type, each containing values associated with attributes defined by the object type.

The units are the *unique* values in a dataset that are expected to contribute to the latent semantic structure within the dataset. In general, descriptive values with repetitions within the dataset are well suited as units, in contrast to identifying attributes such as names or key attributes in general.

LSM requires uniform units to produce statistically meaningful results. Hence, for attributes involving continuous data types, discretisation is to be performed to generate a set of intervals as attributes. However, the separation of a Boolean type attribute in its unique values *true* and *false* is not necessary (see 5.6.2 for exemplification).

The objects are thus binary compositions of unique values. To adjust the weight of a particular attribute, all of its unique values have to be weighted in a uniform manner.

To analyse more than one dataset, reverse normalisation of the database has to be performed to produce one dataset. However, for this purpose we suggest to use the M-LSM approach described in 5.2.

The peculiarity of LSDM is thus the general method we have proposed for the mapping of a dataset to an occurrence matrix suitable for LSM. Aspects of this method have been used in [31] for semantically enhanced collaborative filtering (see 5.6.3).

## 5.5. Reasoning

The first part of this section provides a brief introduction to *deterministic reasoning* in Description Logics. LSM of ontologies can be considered as *probabilistic reasoning*, and, in the second part of this section, we propose methods for that.

### 5.5.1. Ontology

The Semantic Web models knowledge in an explicit way using ontologies. An *ontology* is a conceptual model of a domain of interest that captures and makes explicit the vocabulary used in semantic applications.

#### Description Logics

*Description Logics* (DL) are a family of formal languages for representing terminological knowledge and for providing a way to reason about this knowledge [1]. DL are the formalism behind an ontology, defining the two component types an ontology consists of, a TBox and an ABox:

A *TBox* is a *terminological* component that describes a domain with *concepts* or *classes* (sets of individuals) and *roles* or *properties* (properties or binary predicates representing links between individuals).

An *ABox* is an *assertional* component that specifies the membership of individuals (objects) or pairs of individuals in concepts and roles, respectively.

### 5.5.2. Reasoning in Description Logics

A DL *reasoner* is able to infer implicit knowledge from the knowledge explicitly contained in a knowledge base such as an ontology, providing a clear distinction and advantage over database systems [1].

### TBox Reasoning Tasks

In the following, the reasoning tasks [1] for TBoxes are listed. Satisfiability is the key reasoning task, all other inferences can be reduced to.

*Satisfiability*: A concept $C$ is satisfiable with respect to the TBox, if there exists a model of that TBox such that $C$ in this model is not empty. Satisfiability thus checks whether a concept makes sense or whether it is contradictory.

*Subsumption*: A concept $C$ is subsumed by a concept $D$ if in every model of the TBox the set defined by $D$ is a subset of the set defined by $C$. Subsumption thus checks if one concept is more general than another.

*Equivalence*: Two concepts $C$ and $D$ are equivalent if in every model of the TBox these concepts are equal.

*Disjointness*: Two concepts $C$ and $D$ are disjoint if in every model of the TBox the intersection of these concepts is empty.

### TBox Classification

Based on the subsumption relationships of the concepts of a TBox, a DL reasoner is able to compute a taxonomy. A *taxonomy* is the hierarchy of concepts. It provides an intuitive way for a human to explore the concepts and their relationships.

### ABox Reasoning Tasks

The reasoning tasks [1] for ABoxes are listed below. Similar to the TBox reasoning tasks, all other reasoning tasks can be reduced to ABox satisfiability.

*ABox satisfiability* checks the assertions in an ABox for satisfiability with respect to the referenced TBox.

*ABox subsumption* checks if a specific object is an instance of the concept $C$ or if this concept subsumes the object.

*ABox realisation* computes the most specific concept for each object in the ABox it is an instance of with respect to the TBox.

### 5.5.3. Probabilistic Reasoning

Based on general methods for LSM of ontologies we have proposed in graph analysis (see 5.3) and LSDM (see 5.4), in this section we propose approaches for probabilistic reasoning.

#### Value Prediction for ABox Realisation

LSDM (see 5.4) can be used on the assertional component of an ontology for prediction purposes, allowing to accomplish the ABox realisation reasoning task with missing or uncertain information.

An automatically retrieved object may contain insufficient information to be realised based on ontological axioms by a DL reasoner. A large amount of retrieved objects is used as an input. Using the information in those objects, LSM can be used to predict missing attribute values (see 5.6.2 for exemplification).

#### Probabilistic Concept Learning

In an ontology, a composition can be a subset of similar classes with the same parent class that are somehow related to another disjoint subset of classes. This would result in the same situation as previously described in graph node clustering in 5.3.2. The clustering result can be interpreted as a suggestion for the introduction of superordinate classes for each group of classes in the set of compositions. As a generalisation of this approach, we propose probabilistic TBox classification in 5.5.3.

#### Probabilistic TBox Classification

In feature-based models of semantic memory concepts are described by unstructured sets of attributes (see 3.4), denoted as semantic features. We propose to use LSM to compute

fuzzy categories in a feature-based model, considering concepts as compositions of unique semantic features.

Similarly, for the global TBox classification we propose to consider concepts as compositions of unique atomic concepts and unique atomic roles, listed implicitly by the right-hand side of axioms. For that, the TBox has to be expanded, and the right-hand side of axioms has to be converted to a uniform representation.

The proposed approach may be considered as a hybrid TBox with the crisp hierarchical structure extended by a fuzzy one, comprising latent relationships among concepts.

Having the fuzzy hierarchy, the approach can be extended to probabilistic ABox realisation. For that, the most specific concept may be determined by querying the reduced occurrence matrix without intervention of a DL reasoner.

## 5.6. Practical Applications

The concept of LSM of ontologies has been successfully applied in practise for collaborative filtering and ontology merging. Those and other examples described in this section illustrate the possible areas of applications for the methods proposed in this chapter.

### 5.6.1. Node Clustering

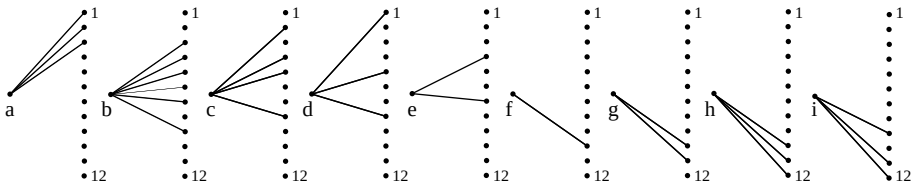



Figure 5.1.: Node clustering

From a fictional graph, the subsets $V_c = \{a, b, ..., i\}$ and $V_u = \{1, 2, ..., 12\}$ were chosen. For each node from the subset of compositions $V_c$ its connections to the subset of units $V_u$ are separately illustrated in Figure 5.1. Each graph in the figure depicts thus a part of the same graph.

Each set of edges corresponding to a node in the subset of compositions can be mapped to a vector, forming the binary occurrence matrix $W$:

$$W = \begin{pmatrix} 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{pmatrix}$$

For the clustering results, see 4.5.4 where virtually the same occurrence matrix have been used. The same procedure can also be used for graphs with weighted edges.

### 5.6.2. Latent Semantic Data Mining

Table 5.2 shows a representation of a fictional dataset of multimedia devices consisting of nine objects. Identifying attributes, such as *model number*, have already been omitted.

#### Feature Extraction

Although all values are numeric, the matrix representation in Table 5.2 is not suitable for LSM, since uniform units are required. For the attributes *display size* and *price* discretisation has to be performed. The attribute *manufacturer* has to be separated in its unique values. The separation of the Boolean type attributes in their unique values *true* and *false* would introduce redundancy and is thus not made.

Table 5.3 shows the resulting occurrence matrix $W$. Every device has an *audio player*, this attribute is thus redundant and is neglected for further analysis.

Table 5.2.: Representation of a dataset unsuitable as occurrence matrix

| Attributes | $c_1$ | $c_2$ | $c_3$ | $c_4$ | $c_5$ | $c_6$ | $c_7$ | $c_8$ | $c_9$ |
|---|---|---|---|---|---|---|---|---|---|
| manufacturer | 1 | 1 | 1 | 2 | 2 | 2 | 3 | 3 | 3 |
| display size | 1.8 | 2.0 | 3.2 | 2.0 | 2.8 | 2.8 | 1.2 | 1.8 | 2.0 |
| picture viewer | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 0 |
| audio player | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| video player | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 |
| price | 78 | 129 | 228 | 99 | 157 | 256 | 56 | 115 | 219 |

Table 5.3.: The occurrence matrix $W$ of the dataset in Table 5.2

| $i$ | Unique Attribute Values | $c_1$ | $c_2$ | $c_3$ | $c_4$ | $c_5$ | $c_6$ | $c_7$ | $c_8$ | $c_9$ |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | manufacturer 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | manufacturer 2 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 |
| 3 | manufacturer 3 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 |
| 4 | small display | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 1 |
| 5 | large display | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 |
| 6 | picture viewer | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 0 |
| - | audio player | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 7 | video player | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 |
| 8 | price < 100 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 |
| 9 | $100 \geq$ price $\geq 200$ | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 |
| 10 | price > 200 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 |

#### Value Prediction

Missing or distorted values can be considered as obscuring noise, which is removed through dimensionality reduction. Hence, value prediction is simply a different interpretation of the latent semantic structure.

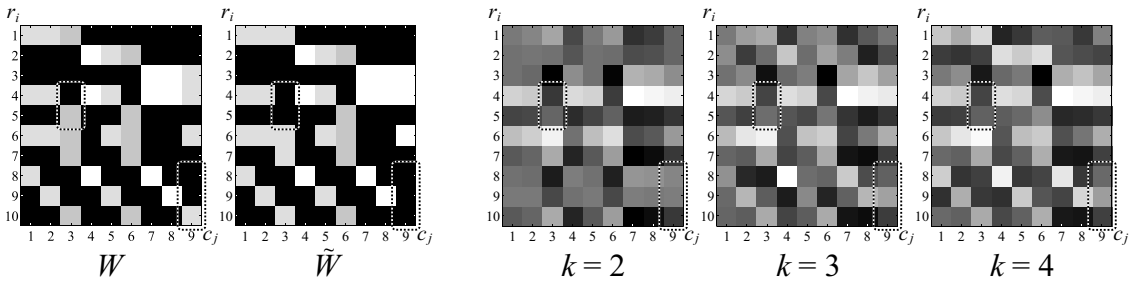



Figure 5.2.: Value prediction for probabilistic reasoning

For the demonstration, the values for *display size* of $c_3$ and *price* of $c_9$ are omitted. This may happen, when a dataset have been automatically extracted from a document. The original occurrence matrix $W$ and the altered version $\tilde{W}$ along with the low rank approximation of $\tilde{W}$ for different values of $k$ is visualised in Figure 5.2. The unique values of the omitted attributes are encircled.

For *display size* of $c_3$ the most probable value is *large display*. This prediction matches the original value.

For the *price* of $c_9$ the most probable value is $100 \geq price \geq 200$, which can be determined for $k = 3$ and $k = 4$. The prediction does not match the original value $price > 200$. However, the lower price range predicted is an indication for a bad price/performance ratio compared to other devices, possibly justified by *small display* and the absence of *picture viewer*.

### 5.6.3. Collaborative Filtering

*Collaborative filtering* (CF) is a technology used by recommender systems to combine opinions and preferences of users in a community in order to achieve personalised recommendations. Table 5.4 illustrates simple examples of LSM applied in CF.

Table 5.4.: Examples for Latent Semantic Mapping in collaborative filtering

| Application | Units | Compositions |
|---|---|---|
| Multimedia Organiser | title | playlist |
| Online Shop | article | cart |
| Lecture Scheduler | lectures | schedule |
| Semantically Enhanced CF (5.6.3) | unique values | items |

#### Semantically Enhanced Collaborative Filtering

In semantically enhanced CF [31], structured semantic knowledge about items is used in conjunction with user-item ratings to create a combined similarity measure for item comparisons. Semantic knowledge is automatically extracted from the Web based on reference domain ontologies. The approach for integrating semantic similarities into the standard item-based CF framework involves performing LSM on the semantic attribute matrix. Then, item similarities are computed based on the reduced semantic attribute matrix, as well as based on the user-item ratings matrix. Finally, the linear combination of those two similarities is used to perform item-based CF.

The semantic attribute matrix is a binary occurrence matrix, where compositions are objects (items) and units are the unique values of the semantic attributes associated with those objects. The attribute values are treated the same way as described in LSDM (see 5.4). For attributes involving a concept hierarchy, each concept node is represented as a unique attribute value.

### 5.6.4. Ontology Merging

Ontologies are conceived as a means of sharing and reusing knowledge. Hence a typical task is to compare several ontologies and to combine them into a more extensive one. This process is known as *ontology merging*.

#### The HCONE Approach to Ontology Merging

Human Centered Ontology Engineering Environment (HCONE) makes use of the intended informal meaning of concepts by mapping them to *WordNet* (see 3.1.3) senses using LSM. Based on these mappings and using the reasoning services of Description Logics, ontologies are automatically aligned and merged [27].

The approach is based on the assumption that ontologies being merged exhibit a meaningful *linguistic* structure regarding the vocabulary used as names of concepts and ontological axioms, restricting to their intended interpretations of those names. The objective of the algorithm is to derive an intermediate ontology from *WordNet* that best matches the vocabulary and axioms of both original ontologies.

The algorithm gets for a concept all senses from *WordNet* lexicalised by the name of the concept. In order to build the occurrence matrix, each sense is considered as composition of terms in the vicinity of the sense. Those terms are the synonyms in the minimal configuration. The vicinity can be extended by hyperonyms and hyponyms and senses of each.

A query is constructed from names of concepts in the vicinity of the analysed concept. In the minimal configuration those are primitive parents and children. Taxonomy parents and concepts related via domain specific relations can be optionally added. With that query, the *WordNet* sense that best matches the concept is retrieved from the reduced occurrence matrix.

Based on the mappings to the intermediate ontology a reasoner finally aligns and merges the original ontologies automatically.

## 5.7. Summary

Originally developed in the context of information retrieval, LSA has been successfully applied in many other areas, due to the generic properties it exhibits: the mapping of discrete entities onto a continuous parameter space, the dimensionality reduction, and the global outlook.

The approach is generalised to LSM, suitable for any discrete data, consisting of objects of two related object types. The corresponding objects are denoted as units and as compositions of those units. The further generalisation M-LSM allows to analyse data, consisting of objects of more than two interrelated object types.

For the application of LSM on databases we have introduced the novel term Latent Semantic Data Mining (LSDM) to convey the relation of LSM to data mining. The peculiarity of LSDM is the general method we have proposed for the mapping of a dataset to an occurrence matrix suitable for LSM. LSDM can be used for object clustering and value prediction. Based on LSDM and graph analysis, we have proposed approaches for probabilistic reasoning.

# 6. Discussion

This chapter provides an interpretative summary of the thesis, discussing and evaluating the covered material and the achieved results.

## 6.1. Human Memory

The structure of the human memory suggests that the semantic memory is the only type of human memory that can be communicated and thus acquired by a computer. The DIKW hierarchy and the more abstract definitions of the terms data, information, knowledge and wisdom suggest that information technology has to evolve towards knowledge technology to achieve satisfying results in natural language processing. For a computer, concepts represented by words remain abstract due to the absence of relations to real life experience (stored in the episodic memory) and skills (stored in the procedural memory). The structure of the human memory suggests that semantic memory is the only memory that can be actually communicated. It is thus the only type of human memory a computer can potentially acquire from text documents collected on the Web. A machine can never acquire full human knowledge without having a human body to experience life, without being human. Furthermore, machines can never reach wisdom, which gives people the ability to create future. The implication is that machines will always remain nothing more than sophisticated tools, helping humans to achieve their goals.

## 6.2. Semantic Memory Models

Knowledge-based technologies use computational cognitive semantic memory models. In contrast to cognitive systems, the semantic memory models are required to have a manageable computational complexity rather than to simulate a full range of human cognitive abilities. The memory models can be classified into four types: network, statistical, feature-based and associative.

The Semantic Web models knowledge in an explicit way using ontologies, hierarchical networks of interrelated concepts. An ontology is a network model of the semantic memory. Latent Semantic Analysis (LSA) models semantic knowledge in an implicit way by mapping documents to a continuous vector space and reducing the dimensionality of the data. Document retrieval and comparison are performed on the data with reduced dimensionality, allowing to overcome the problem of word sense disambiguation. LSA has also demonstrated the ability to model human semantic memory by learning from a large corpus of representative English text.

All semantic memory models overlap in their basis; they all describe relationships between corresponding information pieces. Each model thus can be represented by a graph or a matrix, allowing a variety of mathematical methods to be applied for analysis. Statistical and associative models use a matrix directly for representing knowledge. Feature-based models can be represented by a matrix. A network model can be converted to a feature-based model in order to be represented by a matrix. The idea of the application of LSA on ontologies thus implies finding a matrix representation for an ontology, and reducing its rank afterwards to uncover a latent semantic structure in the ontology.

## 6.3. Latent Semantic Analysis

The challenge is to find a suitable matrix representation for an ontology. Therefore the principles of LSA have been investigated. LSA uses singular value decomposition (SVD) to reduce the dimensionality of the original data to uncover the latent semantic structure. This structure is determined by global correlation patterns caused by the term co-occurrence among documents. This approach is known as principle component analysis (PCA). The visual explanation of LSA by means of greyscale images, dendrograms and distance graphs has helped to better understand how the matrix is factorised by the singular value decomposition and what the effects of the dimension reduction are. The dimension reduction causes related documents to move closer together in the vector space (decrease of the average intra-topic distance), while the distance between groups of related objects (inter-topic distance) remains largely unaffected. The span between average intra-topic and inter-topic distances grows with the decreasing number of retained dimensions.

## 6.4. Latent Semantic Mapping of Ontologies

LSA have been successfully applied in natural language processing and other areas. Three specific factors seem to make LSA particularly attractive: the mapping of discrete entities to a continuous parameter space; the dimensionality reduction inherent in the process; and the intrinsically global outlook of the approach.

Latent Semantic Mapping (LSM) is a generalisation of LSA suitable for any discrete data, consisting of objects of two related object types. The further generalisation M-LSM allows to analyse data, consisting of objects of more than two interrelated object types.

A feature extraction function used to map the discrete data to a high-dimensional continuous vector space must uniformly reflect statistical properties of the data. In LSA, a function of the term frequency, such as TF-IDF, is used for the feature extraction. In graph analysis the binary adjacency matrix can be directly used as occurrence matrix.

For the mapping of datasets, consisting of attributes of different data types, we have proposed to consider objects as binary compositions of unique attribute values. LSM can be considered as a method for probabilistic data mining in this context. To convey this, we have introduced the novel term Latent Semantic Data Mining (LSDM) for LSM of databases.

LSDM and graph analysis in conjunction provide general principles for LSM of ontologies, relinquishing an ontology expert to find a specific practical application by recognising the data suitable for this analysis. Based on those principles, we have proposed approaches for probabilistic reasoning, enabling A-Box realisation by a DL reasoner based on predicted values, and probabilistic concept learning.

As a generalisation of the probabilistic concept learning, we have proposed probabilistic TBox classification, which may allow probabilistic A-Box realisation without intervention of a DL reasoner. However, this is a subject of further research.

# 7. Conclusion

To conclude the thesis, this chapter summarises the achieved results. Afterwards, we provide ideas for future work.

## 7.1. Results

For Latent Semantic Mapping of ontologies we have proposed methods for graph analysis and data mining in relational databases. Based on these methods, approaches for probabilistic reasoning have been derived, satisfying the main objective of this thesis. As approaches for probabilistic reasoning we have proposed value prediction for ABox realisation, concept learning and probabilistic TBox classification.

To visualise the functioning principle of LSA we have proposed three techniques: greyscale image, dendrogram and distance graph. The visual explanation helps to better understand how the matrix is factorised by the singular value decomposition and what the effects of the dimension reduction are. Important effect of the dimension reduction is the increasing span between average intra- and inter-cluster distances with decreasing number of retained dimensions.

LSA and the Semantic Web appear quite unrelated at first sight. Psychological foundations of knowledge modelling have yielded revealing relationship between LSA and ontologies; both can be considered as models of the human semantic memory. The covered material also provides a basis to understand the potential and the limitations of the upcoming knowledge-based technologies such as the Semantic Web.

## 7.2. Future Work

### 7.2.1. Rank Estimation

There is no general procedure known for choosing $k$, the number of retained dimensions; it is rather an empirical issue and depends on methods used for the evaluation of the retrieval results. Based on the observation that with decreasing $k$, the span between intra- and inter-cluster distances increases, we have proposed a novel approach for choosing $k$, depending on that span.

This is, however, not a trivial task, since not every document collection can be separated in meaningful topics. This is particularly the case when all documents in the collection are on the same topic. Furthermore, it is unclear, how the span is related to the retrieval performance. It is thus unclear, if some optimal $k$ regarding the span also guarantee a better retrieval performance.

### 7.2.2. Probabilistic TBox Classification

Probabilistic TBox classification we have proposed may allow probabilistic A-Box realisation without an intervention of a DL reasoner.

For the mapping of an entire TBox to a feature-based model, to derive an occurrence matrix from, we have proposed to consider complex concepts as compositions of atomic concepts and atomic roles as a general approach. The development of a methodology for the feature-based representation of a TBox is a subject of further research.

# A. Software Tools

In this chapter, the software tools have been used during the development of this thesis are briefly described. All trademarks and registered trademarks are the property of their rightful owners.

## A.1. MATLAB

*MATLAB* is a high-level programming language and interactive environment for numerical computations. *MATLAB* consists of toolboxes, containing ready to use functions for matrix operations, such as `svd` for the singular value decomposition.

We have used *MATLAB 7.5* for the experiments on Latent Semantic Analysis, and to generate the greyscale images and dendrograms.

▷ `http://www.mathworks.com/matlab`

## A.2. GraphViz

*GraphViz* is open source software for visualisation of graphs specified in DOT language script. DOT is a simple plain text graph description language.

Listing A.1: Example DOT script describing a distance graph

```
graph dg {
  node  [shape=none, width=0.05, height=0.05, fontsize=12]
  edge  [penwidth=0.0]
  graph [size=5]

  1 -- 2 [len=1.332855]
  1 -- 3 [len=1.277954]
  1 -- 4 [len=1.332855]
  2 -- 3 [len=1.150262]
```

```
  2 -- 4 [len=1.230959]
  2 -- 5 [len=0.785398]
  2 -- 9 [len=1.332855]
  3 -- 4 [len=0.911738]
  3 -- 5 [len=1.277954]
  6 -- 7 [len=0.785398]
  6 -- 8 [len=0.955317]
  7 -- 8 [len=0.615480]
  7 -- 9 [len=1.150262]
  8 -- 9 [len=0.841069]
}
```

We have used the *GraphViz* layout program *neato* to generate the distance graphs in Figure 4.9 from the output generated in *MATLAB*. As an example, the description of the distance graph for $k = 9$ is shown in Listing A.1. *ZGRViewer* have been used as a graphical user interface for *GraphViz*.

▷ `http://www.graphviz.org`

## A.3. Snowball

*Snowball* is a string processing language designed for creating stemming algorithms for use in Information Retrieval.

We have used the stopword list and the stemming algorithm from *Tartarus Snowball* to produce the demonstration of the document preprocessing in Figure 4.1.

▷ `http://snowball.tartarus.org`

## A.4. CorelDRAW

*CorelDRAW* is a vector graphics editor. It has the ability to import SVG, EPS and PDF among many other vector graphics formats for full editing. The result can be directly exported to PDF.

We have used *CorelDRAW X3* to produce all diagrams in this thesis, and figures assembled from the outputs generated in *MATLAB* and *GraphViz*.

▷ http://www.corel.com/coreldraw

## A.5. LaTeX

*LaTeX* is a document preparation system for the for the typesetting engine *TeX*. It includes features designed for the production of technical and scientific documentation. In contrast to word processors, *LaTeX* encourages authors not to worry too much about the appearance of their documents but to concentrate on getting the right content.

For the composition of this thesis, we have used *MiKTeX 2.6*, an implementation of *TeX* and *LaTeX*, and other related programs for the *Windows* operating system.

▷ http://www.miktex.org

As an environment for document development we have used *LEd*. Among other functions, it offers descriptive hints for *LaTeX* commands, code completion mechanism, word wrapping and code folding.

▷ http://www.latexeditor.org

# B. Source Code

In this chapter we provide code snippets for *MATLAB* for the reproduction of the experiments on Latent Semantic Analysis.

## B.1. Input Data

### B.1.1. Text Documents

Listing B.1: Reading documents from files given sequential numbers as names

```
n = 1;
Documents = cell(0);
while 1
  fid = fopen([int2str(n) '.txt'], 'r');
  if (fid == -1), break, end % no more files found
  n = n + 1;
  Documents{n} = (fread(fid, '*char'))';
  fclose(fid);
end
```

Listing B.2: Read terms from a text file

```
Terms = textread('terms.txt', '%s');
m = size(T,1);
```

#### Feature Extraction

Listing B.3: Compute the occurrence matrix with raw term frequencies

```
A = zeros(m, n);
for d = 1:n
 for t = 1:m
   A(i, j) = size(findstr(Documents{i}, Terms{j}), 2);
 end
end
```

Listing B.4: Normalised term frequency (4.4) with the first norm (4.5)

```
for j = 1:n
  A(:, j) = A(:, j) / sum(A(:, j));
end
```

Listing B.5: Normalised term frequency(4.4) with the second norm (4.6)

```
for j = 1:n
  A(:, j) = A(:, j) / norm(A(:, j));
end
```

Listing B.6: Term frequency – inverse document frequency (4.7)

```
idf = log(n ./ sum(sign(A')))';
for j = 1:n
  A(:, j) = A(:, j) .* idf;
end;
```

### B.1.2. Images and Noise

Listing B.7: Input matrices for the rank estimation experiment (see 4.2.2)

```
A = imread('input_a.bmp');
A = im2double(A(:,:,1));
m = size(A, 1);
n = size(A, 2);

N_B = im2bw(rand(m,n));
N_C = rand(m,n);
p = 0.02;
N_D = (max((1-p), rand(m,n)) - (1-p) * ones(m,n)) / p;

B = double(A & N_B);
C = min(B, N_C);
D = max(C, N_D);
```

## B.2. Document Retrieval

Listing B.8: Document retrieval

```
r = (q'*Ak)';
[sr, id] = sort(r);
```

```
fprintf(1, '\nsearch request: "%s"\n\n', Terms{search_request});
for j = n:-1:1
 fprintf(1, '%5.2f  %s\n', sr(j), Documents{id(j)})
end
```

## B.3. Dimension Reduction

Listing B.9: Dimension reduction (4.20)

```
[U,S,V] = svd(A);
Sk = S .* [ones(m, k) zeros(m, n-k)]; % retain k singular values
Ak = U * Sk * V';
```

## B.4. Distance Matrices

Listing B.10: Distance matrices (4.16), (4.14), (4.26), (4.17)

```
T = M; % term document matrix
for j = 1:n
  T(:, j) = T(:, j) / norm(T(:, j));
end

D_angle = T' * T;
D_angle_threshold = (D_angle > cos(pi/4));
D_correlation = corrcoef(M);
```

## B.5. Visualisation

Listing B.11: Greyscale image

```
colormap(gray);
imagesc(M); % display the image
% pcolor(M); % an alternative to imagesc
% spy(M); % an alternative to imagesc suitable for binary matrices
colorbar('vert'); % display colour bar
axis image; % keep aspect ratio
```

Listing B.12: Single linkage clustering and dendrogram

```
X = M';
Y = pdist(X, 'cosine'); % angle cosine distances
Z = linkage(Y, 'single'); % single linkage clustering

dendrogram(Z, 'orientation', 'right');
```

Listing B.13: Distance graph output for *GraphViz*

```
fid = fopen('output.dot', 'w');

fprintf(fid, 'graph dg {\n');
fprintf(fid, '  node [shape=none, width=0.05, height=0.05, fontsize
   =12]\n');
fprintf(fid, '  edge [penwidth=0.0]\n');
fprintf(fid, '  graph [size=5]\n\n\n');

min_dist = 0.1; % minimum angle distance
max_dist = 0.001; % maximum angle cosine distance

fprintf(fid, '\n\n')
for j = 1:n
 for jj = (d + 1):n
   dist = D_angle(j, jj);

   if (dist > max_dist)
    dist = acos(dist);

    if (dist < min_dist)
      dist = min_dist;
    end;

    fprintf(fid, '  %d -- %d [len=%f]\n', j, jj, dist);
   end
 end
end

fprintf(fid, '}\n');
fclose(fid);
```

# Bibliography


[1] The description logic handbook: Theory, implementation, and applications. 2003.

[2] The semantic web: Research and applications, 4th european semantic web conference, eswc 2007. 5021, 2008.

[3] Wikipedia, the free encyclopedia. 2008.

[4] R. L. Ackoff. From data to wisdom. *Journal of Applied Systems Analysis*, 16, 1989.

[5] Syed Ahsan and Abad Shah. Data, information, knowledge, wisdom: A doubly linked chain? *University of Engineering and Technology, Lahore*, 2006.

[6] J. R. Anderson. *Language, Memory and Thought.* Mahwah, NJ: Erlbaum, 1976.

[7] R. C. Atkinson and R. M. Shiffrin. Human memory: A proposed system and its control processes. *In K. W. Spence and J. T. Spence, The Psychology of learning and motivation: Advances in research and theory*, 2:89 – 105, 1968.

[8] Alan D. Baddeley. *Essentials of Human Memory*. Psychology Press, Taylor and Francis, 1999.

[9] Jerome R. Bellegarda. Exploiting latent semantic information in statistical language modeling. *Proceedings of the IEEE*, 88(8):1279 – 1296, 2000.

[10] Jerome R. Bellegarda. Latent semantic mapping: dimensionality reduction via globally optimal continuous parameter modeling. *IEEE Workshop on Automatic Speech Recognition and Understanding*, pages 127 – 132, 2005.

[11] Jerome R. Bellegarda. *Latent Semantic Mapping: Principles And Applications*. Morgan and Claypool Publishers, 2008.

[12] Tim Berners-Lee, James Hendler, and Ora Lassila. The semantic web. *Scientific American*, 2001.

[13] Michael W. Berry, Zlatko Drmac, and Elizabeth R. Jessup. Matrices, vector spaces, and information retrieval. *SIAM Rev.*, 41(2):335–362, 1999.

[14] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.

[15] K. K. Breitman, M.A. Casanova, and W. Truszkowski. *Semantic Web: Concepts, Technologies and Applications*. Springer-Verlag London Limited, 2007.

[16] Scott Deerwester, Susan T. Dumais, George W. Furnas, Thomas K. Landauer, and Richard Harshman. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41(6):391–407, 1990.

[17] Yihong Gong and Xin Liu. Generic text summarization using relevance measure and latent semantic analysis. pages 19 – 25, 2001.

[18] David J. Hand, D. J. Hand, and Heikki Mannila. *Principles of Data Mining (Adaptive Computation and Machine Learning)*. The MIT Press, 2001.

[19] Peter Hildebrandt. Vom verborgenen sinn. 2007.

[20] Thomas Hofmann. Probabilistic latent semantic analysis. In *UAI*, 1999.

[21] J. Hollan. Features and semantic memory: Set-theoretical or network model? *Psychological Review*, 82:154 – 155, 1975.

[22] Michael Huggett, Holger Hoos, and Ron Rensink. Cognitive principles for information management: The principles of mnemonic associative knowledge (p-mak). *Minds and Machines*, 17(4):445–485, 2007.

[23] Richard D. Hull, Eugene M. Fluder, Suresh B. Singh, Robert B. Nachbar, Simon K. Kearsley, and Robert P. Sheridan. Latent semantic structure indexing (lassi) and comparison to toposim. *Journal of Medicinal Chemistry*, 44:1185 – 1191, 2001.

[24] I. T. Jolliffe. *Principal Component Analysis*. Springer, 2002.

[25] W. Kintsch. The role of knowledge in discourse comprehension: A construction-integration model. *Psychological Review 95*, pages 163 – 182, 1988.

[26] Roberta L. Klatzky. *Human memory: structures and processes (Second Edition)*. San Francisco: Freeman, 1980.

[27] Konstantinos Kotis and A. Vouros. Human-centered ontology engineering: The hcome methodology. *Knowl. Inf. Syst.*, 10(1):109–131, 2006.

[28] T. K. Landauer, P. W. Foltz, and D. Laham. An Introduction to Latent Semantic Analysis. *Discourse Processes*, 25:259–284, 1998.

[29] The MathWorks, Inc. *MATLAB Documentation*, 2007.

[30] M. McCloskey and S. Glucksberg. Natural categories: Well defined or fuzzy sets? *Memory and Cognition*, 6:462 – 472, 1978.

[31] Bamshad Mobasher, Xin Jin, and Yanzan Zhou. Semantically enhanced collaborative filtering on the web. 2004.

[32] Thomas Nagel. What is it like to be a bat? *Philosophical Review*, 83:435 – 50, 1974.

[33] David J. Newman and Sharon Block. Probabilistic topic decomposition of an eighteenth-century american newspaper. 2006.

[34] Hector Oscar Nigro, Sandra Elizabeth Gonzalez Cisaro, and Daniel Hugo Xodo, editors. *Data Mining With Ontologies: Implementations, Findings and Frameworks*. Idea Group Reference, New York, 2007.

[35] Rifat Ozcan and Y. Alp Aslandogan. Concept based information access using ontologies and latent semantic analysis. Technical report, 2004.

[36] J. G. W. Raaijmakers and R. M. Shiffrin. Search of associative memory. *Psychological Review*, 88:93 – 134, 1981.

[37] Jennifer Rowley. The wisdom hierarchy: representations of the dikw hierarchy. *Journal of Information Science*, 33(2), 2007.

[38] G. Salton, A. Wong, and C. S. Yang. A vector space model for automatic indexing. *Commun. ACM*, 18(11):613 – 620, 1975.

[39] R. Sibson. Slink: An optimally efficient algorithm for the single-link cluster method. *The Computer Journal*, 16(1):30 – 34, 1973.

[40] E. E. Smith, E. J. Shoben, and L. J. Rips. Structure and process in semantic memory: A featural model for semantic decisions. *Psychological Review*, 81:214 – 241, 1974.

[41] Daniel A. Spielman and Shang Teng. Spectral partitioning works: Planar graphs and finite element meshes. Technical report, Berkeley, CA, USA, 1996.

[42] Xuanhui Wang, Jian-Tao Sun, Zheng Chen, and ChengXiang Zhai. Latent semantic analysis for multiple-type interrelated data objects. pages 236–243, 2006.

# List of Figures

# Index