



## Document Control

**Title:** State of the Art Report in Knowledge Sharing, Recommendation and Latent Semantic Analysis

**Author/Editor:** Tereza Iofciu; Xuan Zhou; Bas Giesbers; Ellen Rusman; Jan M. van Bruggen; Stefano Ceri

**E-mail:** [iofcu@l3s.de](mailto:iofcu@l3s.de); [xuan@l3s.de](mailto:xuan@l3s.de); [Bas.giesbers@ou.nl](mailto:Bas.giesbers@ou.nl); [Ellen.rusman@ou.nl](mailto:Ellen.rusman@ou.nl); [Jan.vanbruggen@ou.nl](mailto:Jan.vanbruggen@ou.nl); [ceri@elet.polimi.it](mailto:ceri@elet.polimi.it)

## Amendment History

Version	Date	Author/Editor	Description/Comments
1	Feb 2006	1	First Draft of L3S literature review
2	March 2006	3	Outline of state of the art on Latent Semantic Analysis
3	March 2006	3	First draft of state of the art on LSA
4	April 2006	3	Second draft of state of the art on LSA
5	May 2006	3	Third draft of state of the art on LSA, after a review of the second draft by Stefan Trausan-Matu.
6	May 2006	1	First Version of state of the art on recommendation systems
7	May 2006	1	Deliverable outline
8	May 2006	1	Deliverable draft
9	May 2006	1	Deliverable first version
10	May 2006	1,3	Deliverable, after review from Stefano Ceri

## **Legal Notices**

The information in this document is subject to change without notice.

The Members of the COOPER Consortium make no warranty of any kind with regard to this document, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. The Members of the COOPER Consortium shall not be held liable for errors contained herein or direct, indirect, special, incidental or consequential damages in connection with the furnishing, performance, or use of this material.

## **Guidelines for Completing the Deliverable Reporting Template:**

**Font: Arial;                      Font Size: 11;                      1,2 line-spacing**

Please include all necessary information relating to the completion of this Deliverable. Attach relevant materials as necessary (copies of publications; course and/or conference programs, etc.). Elements that should be incorporated into this report include:

- Public Events (Workshops, Conferences etc.)
- Integration Activities (Research Exchange, Scholarships & Travel Grants)
- Publications (Articles, Papers, Press releases etc.)
- Abstract
- Index

<b>COLLABORATIVE OPEN ENVIRONMENT FOR PROJECT-CENTRED LEARNING</b>	<b>1</b>
<b>EUROPEAN COMMISSION SIXTH FRAMEWORK PROJECT (IST-027073)</b>	<b>1</b>
<b>AMENDMENT HISTORY</b>	<b>2</b>
<b>LEGAL NOTICES</b>	<b>2</b>
<b>1 EXECUTIVE SUMMARY</b>	<b>6</b>
<b>2 RECOMMENDER SYSTEMS</b>	<b>7</b>
<b>2.1 Introduction to Recommender Systems</b>	<b>7</b>
<b>2.2 State of the Art on Recommender Systems</b>	<b>8</b>
2.2.1 Content-based recommendation	8
2.2.2 Collaborative recommendations	9
2.2.3 Other recommender techniques	11
2.2.4 Hybrid approaches: combinations of different recommendation techniques	12
2.2.5 Semantically Rich Recommendations in Social Networks	13
<b>2.3 Recommendation Services in COOPER</b>	<b>13</b>
<b>3 LATENT SEMANTIC ANALYSIS</b>	<b>15</b>
<b>3.1 Introduction</b>	<b>15</b>
3.1.1 Definition	15
3.1.2 LSA in a nutshell	17
<b>3.2 Application areas of LSA</b>	<b>18</b>
3.2.1 Document retrieval and latent semantic indexing (LSI)	18
3.2.2 Representation of semantics and discourse processing	20
3.2.3 Educational applications	22
3.2.4 Human Resource Management	26
<b>3.3 Implementation issues: corpus construction</b>	<b>27</b>
3.3.1 Corpus size	27
3.3.2 Document selection	28
3.3.3 Document size	29

3.3.4	Stemming	29
3.3.5	Stopping	29
<b>3.4</b>	<b>Evaluation</b>	<b>29</b>
3.4.1	Strengths of LSA	29
3.4.2	Weaknesses of LSA	30
<b>3.5</b>	<b>Relevance of LSA for Cooper</b>	<b>31</b>
3.5.1	Community formation, community support and collaboration	31
3.5.2	Human Resource Management and task allocation	32
3.5.3	Localizing resources	32
3.5.4	Support of assessment and feedback	32
<b>4</b>	<b>KNOWLEDGE SHARING SYSTEMS</b>	<b>33</b>
<b>4.1</b>	<b>Knowledge Sharing Applications on the Web</b>	<b>33</b>
4.1.1	Discussion Forums	33
4.1.2	Blogs and RSS feeds	34
4.1.3	Wikis	34
4.1.4	Folksonomies	35
<b>4.2</b>	<b>Connection to Cooper</b>	<b>35</b>
<b>5</b>	<b>CONCLUSIONS</b>	<b>37</b>
<b>6</b>	<b>BIBLIOGRAPHY</b>	<b>38</b>

# 1 Executive Summary

The COOPER scenario is characterized by a virtual team of persons with heterogeneous backgrounds and competencies, who are geographically dispersed and work together in projects to solve complex problems. The objective of COOPER is to provide an environment that can not only coordinate the collaboration within the virtual teams, but also support them with learning facilities, so that they can obtain the necessary knowledge and competencies to drive their projects to final success. Knowledge sharing, which plays an essential role in all collaborative learning environments, is one of the most important facilities in the COOPER environment.

COOPER supports knowledge sharing within a project team, between concurrent projects, as well as across current projects and historical projects. To enable such knowledge sharing, the following basic instruments have to be developed or used: 1) tools that enable users to input or to co-construct knowledge, e.g. discussion forums and Wiki; 2) knowledge repositories that archive and manage the learning materials used in projects, the knowledge items produced during the projects (e.g. deliverables, entries in the Project Wikis) and various external knowledge resources brought in by users; 3) tools for searching and browsing the repository.

While these instruments form a fundamental framework for knowledge sharing, they do not ensure that users will be able to share knowledge effectively. Besides this basic framework, the COOPER platform should be equipped with more advanced services to enforce efficient knowledge sharing. For example, a user may not be aware of the knowledge he needs or may not be able to tell what learning materials are suitable for his knowledge level, so it will take him much time or effort to find the right learning step to take. Thus, the COOPER platform should be able to give users appropriate advices on suitable knowledge resources based on their profiles and current activities. On the other hand, a good knowledge sharing environment should be able to establish networks among users, so as to foster a social climate required by knowledge sharing. This is especially important for COOPER, whose users usually have heterogeneous backgrounds and expertises.

The WP3 of COOPER is supposed to create a knowledge sharing framework and complement it with various services necessary for project-centred knowledge sharing. The technologies of *recommender systems* and *latent semantic analysis (LSA)* will play important roles in this work package. The former enables the COOPER platform to assess users' knowledge needs, and recommend them relevant learning materials or experts to communicate with. The latter enables the COOPER platform to understand the background and expertise of each user, so as to establish the connections among them.

In this deliverable, we review the state of the art of the relevant technologies that will be utilized to facilitate the knowledge sharing in COOPER. In Section 2, we introduce the current technology of recommender systems and give several scenarios in COOPER that need recommendation services. In Section 3, we give an overview of latent semantic analysis (LSA) and its various applications, and show how it could be utilized to help knowledge sharing. In Section 4, we survey the most popular web tools for knowledge sharing and knowledge co-construction.

## 2 Recommender Systems

To achieve effective knowledge sharing in COOPER, it is not enough only to provide a knowledge repository for users to input and search for knowledge items. Instead, the COOPER environment should be able to assess users' knowledge needs and actively advise users on useful learning materials, co-workers with relevant expertises and the means to obtain these resources. In this section, we review the current technologies of recommender systems and envision how they could be utilized in the COOPER scenario.

Section 2.1 gives an introduction of recommender system and its various applications. Section 2.2 summarizes the most widely used techniques of recommender systems. Section 2.3 shows how the recommendation techniques could be used in COOPER to facilitate knowledge sharing and the management of virtual teams

### 2.1 Introduction to Recommender Systems

Recommender systems are widely used in today's web applications to provide users with individualized and useful suggestions from a large pool of online products or information entries. The first description of recommender systems appeared in mid-1990's [Resnick et al. 1994; Hill et al. 1995; Shardanand and Maes 1995] where they were originally defined as systems in which "people provide recommendations as inputs, which the system then aggregates and directs to appropriate recipients" [Resnick & Varian 1997]. Since then, recommender systems have become fundamental applications in e-commerce and information access areas.

In the book section of *Amazon.com* ([www.amazon.com](http://www.amazon.com)), the description page of each book contains not only the information about the content and the purchase details but also all kinds of recommendations for other interesting books. For example, the *Customer Who Bought* feature, available on each description page, consists of two types of recommendations. One recommends the books bought by users who bought the selected book, and the other recommends authors whose books are frequently bought by users who bought books by the author of the selected book. The *Book Matcher* feature allows customers to rate the books they have read on a scale from five ("hated it") to one ("loved it"). These ratings will be used by recommender system to estimate users' interests and to give more appropriate recommendations. Users can also give feedback for the books they were recommended.

Similar to *Amazon.com*, *CDNOW* ([www.cdnw.com](http://www.cdnw.com)) has for each music album a description page. Here customers can find the *Customers Who Bought* feature, which presents ten other albums which were purchased by users who bought the selected album. *My CDNOW* feature enables customers to set up their own music store based on what they like. For example, customers can specify what albums they own, which they like and which artists they like. Based on their preferences they can ask for recommendations. They can also specify the albums they want and the recommendations they receive will be adjusted to these specification.

Another example is the *Reel.com*'s *Movie Matches* ([www.reel.com](http://www.reel.com)), which provides recommendations from different perspectives, such as "close matches" and "creative matches".

The technique of recommender system relies on the combination of several technologies from different areas. They include cognitive science, social science, computer technology, statistics and also information management. In principle, a recommendation process starts from the background data including user profiles and the initial set of ratings on existing items, and the input data including the criteria provided by users, and applies an algorithm which combines the background and input data to arrive to suggestions for the user.

## 2.2 State of the Art on Recommender Systems

Basically, a recommender system deals with two entities: *Users* and *Items*. Each element from the *Users* space is associated with a profile with different characteristics; the same with elements from the *Items* space. The utility of an item to a user is usually represented by a *rating function*. The problem of giving recommendation can be abstracted to the problem of estimating ratings for items which have not been seen by the user. The items with the highest estimated ratings are then recommended to the user.

To be specific, the system tries to estimate the rating function  $R$

$$R: \text{Users} \times \text{Items} \rightarrow \text{Ratings}$$

Normally, this function has already been partially defined by users, who have explicitly specified the ratings for a subset of the (*user, item*) domain. The main task of the recommender system is to extrapolate the function from the subset to the whole *Users x Items* space.

Based on the above notions, many researchers consider that recommender systems can be classified as *content-based*, *collaborative* and *hybrid* [Balabanovic and Shoham 1997]. In the rest of this section, we will review the main techniques of recommender systems according to this classification. There also are some other classifications based on *demographic*, *utility-based* and *knowledge-based* approaches which we describe only briefly.

### 2.2.1 Content-based recommendation

Content-based recommendation methods are a continuation of information retrieval [Baeza-Yates and Ribeiro-Neto, 1999; Salton, 1989] and information filtering [Belkin and Croft, 1992] research. Their main technique is based on the “item to item correlation” [Schafer, Konstan and Riedl, 1999]. The system learns the profile of a user through the features of the items the user was previously interested in, and recommends to the user items with similar or related features. Namely, the utility of an item to a user is based on the similarity of the item to the items the user has rated high previously.

Many existing content-based systems focus on recommending items containing textual representation, such as books and news articles. In these systems, each item profile and user profile is usually described with keywords, as it is done in the Fab [Balabanovic and Shoham 1997] and the Syskill and Webert [Pazzani and Billus 1997] systems. In this context keywords are considered to be the most representative words in the documents. The representativeness of a word in a document can be estimated by various statistical measures, in which one of the best known measures is the term frequency/inverse document frequency measure (TF-IDF) [Salton 1989] used in Information Retrieval. Each document is then represented as a vector of the weights of its keywords. (These vectors usually have varying length in different systems.) In order to obtain the ranking of items for a user, the vector of keyword weights of each item is compared against the vector of the user profile, which is generated from the aggregation of the vectors of the items the user was previously interested in. Finally, the items most similar to the user profile are returned as recommendation.

To measure the similarity between vectors of keyword weights, one can resort to different similarity measures. An example is the cosine similarity measure [Salton 1989; Baeza-Yates and Ribeiro-Neto, 1999], which is used widely in information retrieval systems. For example, suppose  $\vec{d}_j$  denotes the vector of a document and  $\vec{p}$  denotes the vector of the user profile, the similarity can be calculated by the *cosine of the angle* between these two vectors:

$$\text{sim}(\vec{d}_j, \vec{p}) = \frac{\vec{d}_j * \vec{p}}{|\vec{d}_j| \times |\vec{p}|} = \frac{\sum_{i=1}^t w_{i,j} \times w_{i,p}}{\sqrt{\sum_{i=1}^t w_{i,j}^2} \times \sqrt{\sum_{i=1}^t w_{i,p}^2}}$$

where  $w_{i,j}$  and  $w_{i,p}$  are the weight of the word  $i$  in the document  $j$  and that in the profile.

In practice, it is not possible to describe each item with a number of most important keywords. For example, in a movie recommender system, the movies are not described by keywords but by attributes like genre, actors, director and so on. In this case, different methods should be applied to measure the similarity between items and user profiles.

Besides the heuristic based approaches from information retrieval, some model based approaches to estimate the rankings have also been used, such as the Bayesian classifiers [Pazzani and Billus 1997]; Mooney et al. 1998] and various machine learning techniques, like clustering, decision trees, and neural networks [Pazzani and Billus 1997]. These methods calculate the estimated ratings based on a *model* learned from the existing data using statistical analysis and machine learning techniques.

The limitations of the content based approach include:

*Limited content analysis.* Content-based recommender systems are limited by the nature of the description of the recommended items. To automate the recommendation process, the description of items has to be represented in a machine-readable format, for instance representing text by a vector of keyword weights. However, limited by current technologies, such machine-readable representation is unable to capture all the important aspects of the objects. For example, the retrieval of keywords from text documents is usually done disregarding of the semantics of the data, such that the system cannot distinguish between a good article and a badly written one when the two articles are described by common keywords. In fact, for text documents information retrieval techniques already work very well. For other domains such as multimedia data, automatic feature extraction is even problematic. This is why it is said that recommender systems have only *limited content analysis* capabilities [Balabanovic and Shoham 1997].

*Over-specialization.* Due to the fact that content-based recommender systems rely solely on the items a user has liked in the past, the user is recommended only items which are similar to the previous ones. When the user has a change of interest, the system will only recommend items similar to his previously registered interests and items similar to the ones already rated high. Ideally, however, the user should be presented with a diversity of items in his interest area and not a homogeneous set. The problem of *over-specialization* is usually limitedly dealt with by introducing a factor of randomness. There are also systems which filter the items which are too different and also too *similar* to the ones the user has already rated, for example the DailyLearner [Billsus and Pazzani 2000].

*New user problem.* In content-based systems recommendations are calculated relative to the items previously rated by the user. A *new user* does not have enough items rated in order to create a rich profile, so that the system is unable to provide adequate recommendations to the user.

## 2.2.2 Collaborative recommendations

The collaborative approach is completely independent of the representation of the recommended items. Instead of recommending to a user the items similar to his previously liked items, this approach recommends the items liked by users with similar preferences (or similar profiles). To make recommendations to a user, the system will first find the user's "peers", who have made similar previous ratings. Then it will try to predict the utility of an item to the user based on the ratings of his "peers" for this item. In other words, the rating  $R(u, i)$  of the item  $i$  for the user  $u$  is estimated based on the ratings  $R(u', i)$  where  $u'$  are users similar to the user  $u$ .

Since the introduction of the collaborative filtering in 1994, there have been many collaborative recommender systems, such as GroupLens and MovieLens [Resnick et al. 1994 ; Konstan et al. 1997], the BellCore Video Recommender [Hill et al. 1995], the Ringo/Firefly [Shardanand and Maer 1995 ] and the Jester system [Goldberg et al. 2001].

The algorithms for collaborative recommendations are of two types: *memory-based* (or heuristic-based) and *model-based* [Bresse et al. 1998].

In memory-based algorithms [Resnick et al. 1994; Shardanand and Maes 1995; Bresse et al. 1998; Nakamura and Abe 1998; Delgado and Ishii 1999], the rating for an item  $i$  for a user  $u$  is computed as the aggregation of the ratings for item  $i$  of the users most similar to the given user. For example the rating  $r_{u,i}$  for user  $u$  and item  $i$  is estimated with an aggregation method like:

$$r_{u,i} = \underset{u \in U_s}{\text{aggr}} r_{u',i}$$

where  $U_s$  is the set of  $n$  users who have rated item  $i$  and are most similar to the user  $u$ , and  $n$  can range from 1 to the total number of users using the system.

The aggregation method is usually a combination of the previous ratings and of a measure of the similarity between users which is used as a weight of the ratings. Some examples of aggregation functions are:

$$r_{u,i} = \frac{1}{n} \sum_{u' \in U_s} r_{u',i}$$

$$r_{u,i} = k \sum_{u' \in U_s} \text{sim}(u, u') \times r_{u',i}$$

$$r_{u,i} = \bar{r}_u + k \sum_{u' \in U_s} \text{sim}(u, u') \times (r_{u',i} - \bar{r}_{u'})$$

where  $k$  is the coefficient used as a normalizing factor, and  $\bar{r}_u$  is the average rating of user  $u$ .

The similarity  $\text{sim}(u, u')$  represents the distance between the two users  $u$  and  $u'$  with respect to their preferences. In existing recommender systems there have been different approaches for measuring the similarity between two users, and most of the approaches consider only the ratings both users have placed. As reported in [Adomavicius et al. 2005], the most popular approaches in recommender systems are the *correlation-based* approach [Resnick et al. 1994; Shardanand and Maes 1995], whose similarity function is:

$$\text{sim}(x, y) = \frac{\sum_{s \in S_{xy}} (r_{x,s} - \bar{r}_x)(r_{y,s} - \bar{r}_y)}{\sqrt{\sum_{s \in S_{xy}} (r_{x,s} - \bar{r}_x)^2 \sum_{s \in S_{xy}} (r_{y,s} - \bar{r}_y)^2}}$$

and the *cosine-based* approach [Breese et al. 1998; Sarwar et al. 2001], whose similarity function is:

$$\text{sim}(x, y) = \cos(X, Y) = \frac{X \cdot Y}{\|X\|_2 \times \|Y\|_2} = \frac{\sum_{s \in S_{xy}} r_{x,s} r_{y,s}}{\sqrt{\sum_{s \in S_{xy}} r_{x,s}^2} \sqrt{\sum_{s \in S_{xy}} r_{y,s}^2}}$$

in which  $S_{xy}$  is the set of all items co-rated by both users  $x$  and  $y$ , and  $r_{x,s}$  and  $r_{y,s}$  are the ratings of item  $s$  assigned by users  $x$ , and  $y$  respectively.

Different techniques have been proposed as extensions to the above collaborative approaches to improve the performance of recommendations. They include *default voting*, *inverse user frequency*, *case amplification* [Breese et al. 1998], and *weighted-majority prediction* [Nakamura and Abe 1998; Delgado and Ishii 1999].

*Model-based* algorithms [Breese et al. 1998; Billsus and Pazzani 1998; Ungar and Foster 1998; Chien and George 1999; Getoor and Sahami 1999; Goldberg et al. 2001] use the collection of previous ratings to learn a *probabilistic model*, which is used afterwards to estimate the other ratings.

Breese et al. 1998 proposed two collaborative filtering approaches based on two probabilistic models: cluster models and Bayesian models, in which ratings are estimated as a probabilistic measure, i.e. the probability that a user would rate an item given his previous ratings. Billsus and Pazzani 1998 proposed a collaborative filtering method in a machine learning framework, in which different machine learning (such as artificial neural networks) techniques and feature extraction (such as singular value decomposition) techniques can be applied. Ungar and Foster 1998 propose a statistical model for collaborative filtering and also compare different methods for estimating the model parameters, such as K-means clustering and Gibbs sampling. Other model-based methods that have been studied include probabilistic relational model and linear regression. Pennock and Horvitz [1999] proposed a method that combines memory-based and model-based approaches, and suggested that the combination of the two methods outperform those using them separately.

The limitations of the content based approach include:

*New user problem.* Similar to the content-based recommender systems, the collaborative systems also give recommendations based on the items previously rated by the user. In particular, the user is compared with other users based on the accumulation of their previous ratings. Therefore, for a *new user* with a few ratings it is difficult to find similar users and the resulting recommendations may be unreliable.

*New item problem.* A *new item* which does not have many ratings is not easily recommended by the system. This problem appears in areas where there is a big stream of new items and each user only rates a few, such as news articles.

*Sparsity.* Collaborative recommender systems rely heavily on the overlap of user ratings and have difficulty when the space of ratings is *sparse*, i.e. few users have rated the same items. One way to overcome this problem is to also use the user profile when calculating user similarity. The sparsity problem can also be weakened by reducing the dimensionality of the item space, in which the comparisons take place. One solution is to use single value decomposition.

Many of the limitations can be reduced by combining content-based and collaborative filtering methods in hybrid recommender systems.

### **2.2.3 Other recommender techniques**

Beside the most popular content-based and collaborative approaches, there are also some other interesting approaches for recommender systems.

*Demographic* recommender systems categorize users based on personal attributes and make recommendations based on demographic information. That is to say, the system collects information about users through demographic studies, and match users against some manually or automatically created stereotypes. Then recommendations are given to each user based on the stereotype he belongs to. The advantage of this recommendation method is that it doesn't need a history of user ratings needed by content-based or collaborative recommender systems. Instead, it forms "people-to-people" correlations using data from demographic studies. Some examples of demographic techniques can be found in Grundy [Rich, 1979], a book recommender system, and the system proposed by Pazzani [1999], which uses machine learning to create a classifier based on demographic data.

*Utility-based* recommender systems base their recommendations on the match between the user's need and the available options that could maximize the expected utility. The main issue

for utility-based recommender systems is how to define the utility function, and existing systems adopt different approaches in computing the utility of items to users. One of the advantages of utility-based techniques is that they allow the introduction of non-product attributes in the utility function, such as price or delivery day, which are regarded important in many e-commerce applications.

*Knowledge-based* recommender systems base their recommendations on inferences about user's needs and preferences. They establish functional knowledge, the knowledge about how the items may meet the user's needs, by case studies or by consulting domain experts. The profile of a user can be looked as the knowledge that supports the inference about his needs. For example, in Google, it may be a simple search query formulated by the user. Shafer, Konsan and Riedl call knowledge-based recommendation the "Editor's choice" method.

#### **2.2.4 Hybrid approaches: combinations of different recommendation techniques**

Hybrid recommender systems (Burke R.) combine two or more recommender techniques in order to gain better performance and to reduce the drawbacks that the individual techniques have. Many hybrid recommender systems combine content and collaborative based approaches, including Fab [Balabanovic and Shahom 1997] and the "collaboration via content" described in Panzzani [1999]. There are different methods for combining the different techniques, as presented below.

*Weighted* recommender systems, where the rating of items is computed as combination, for example linear combination, of the result of the existing recommendation techniques. The P-Tango system [Claypool et al. 1999] uses a linear combination of content and collaborative based recommendations, at first giving them equal weights, and then adjusting the weights according to the feedback given by the user. Panzzani [1999] describes a recommender systems where the outputs of the collaborative, content-based and demographic techniques used are considered to be a set of votes which are then combined in a consensus scheme.

A *switching* recommender system uses some criterion to decide when to switch between the different used techniques. The DailyLearner system uses a hybrid between content-based and collaborative techniques. At first it makes content-based recommendations and when the system seems not to be able to make accurate recommendations it makes collaborative recommendations. As a content-based technique it uses the nearest-neighbour which does not require a large number of examples to give an accurate result, so the ramp-up problem is partially solved. Tran and Cohen [2000] propose a system where the order in which the techniques are applied is not fixed, the order is decided in regard with the past ratings and the recommendations of each technique. Switching hybrids have the disadvantage of introducing additional complexity into the recommendation process due to the new level of parameterization needed for determining the switching criterion.

*Mixed* hybrids present the results from the different recommendation techniques together. The PTV system [Smyth and Cotter 2000] show the results from both content-based and collaborative recommendations. It does not suffer from the new item problem since the content-based component can recommend new shows based on their description. It still has the new user problem as both components need some information about the user in order to give consistent results.

In a *cascade* hybrid system, a recommendation technique is used to produce an initial set of ratings and the second recommendation technique is used to filter this set. This approach allows the system to avoid applying a second lower-priority filter technique on items that are already excluded by the first technique.

A *feature combination* type of hybrid treats the collaborative information as additional feature data associated with each item and applies content-based techniques over the augmented data. With this approach, Basu, Hirsh and Cohen [1998] have reported better results in precision when using a purely collaborative approach. Using all content available features they have shown improvement just in recall and not in precision.

In the *feature augmentation* hybrid systems, one technique is used to obtain a rating or classification and then the information is used as a feature in the process of applying the next recommendation technique. The Libra system [Mooney and Roy 1999] uses data found on Amazon.com and applies content-based techniques on it. The data from Amazon, such as “related authors” and “related titles”, is actually generated using collaborative techniques. The GroupLens team [Sarwar et al. 1998] implemented a set of knowledge-based “filterbots”, such as the number of spelling errors and the size of included messages. These bots were then used as additional information in the collaborative filtering process, acting as artificial users. With this approach they managed to significantly improve the email filtering.

*Meta-level* hybrid systems use the model generated by one technique as input for the second one. In the feature augmentation hybrid the system uses a learned model to generate features as input for the second algorithm, whereas in meta-level hybrids the whole model is used as input. The Fab system [Balabanovic 1997, 1998] is the first meta-level hybrid, and it uses content-based techniques to gather information about the user’s area of interest and then collaborative techniques to gather new pages from the web using all the user models. The advantage of the meta-level hybrid systems is that the learned model is a compressed representation of the user’s profile, and the collaborative filtering that follows can operate on this information easier than on raw data.

### **2.2.5 Semantically Rich Recommendations in Social Networks**

Context information can be very useful in giving recommendations in social networks. For example, when an article is recommended to a user, the context information of the article, including the homepage of its author, the journal where it is published, and other articles that have cited it, could be attached, as it is very likely that the user will need to access such information subsequently. [Ghita S. et. al 2005] explores how context information can be extracted and further used in social networks, and investigates how to make semantically rich recommendations by using the context information.

Context information about resources on the web and on the desktop can sometimes be gathered automatically. For example, context information about publications can be found in the CiteSeer database. It can also be extracted from a user’s browsing actions and interactions with the resources he makes public in the network. In some circumstances, context information needs to be manually defined. The schema defining context information can also specify the transitivity of importance from an object to its contexts, in order to compute how semantically close the context is to the object. The transfer of authority should not be the same in both directions. For example, a paper which cites good publications is not necessarily a good publication, but a paper cited by good publication is of high probability to be a good publication too.

## **2.3 Recommendation Services in COOPER**

There exist a number of scenarios in COOPER that require recommendation services. These scenarios range from the pre-project phase to the project development phases.

In the pre-project phase, the project sponsor selects from a large number of candidate applicants, who have heterogeneous backgrounds, to build a project team, and provides them with academic tutors and project mentors. This process of team building could greatly benefit from appropriate recommendation services, that learn the project objective, its expected results and the technologies involved, and recommend candidate team members or mentors with relevant backgrounds and experiences. Meanwhile, the recommender system can also show statistics of some similar past projects and give hints on what team composition will lead to the promised results. After the team is formed, the system can further study the profiles of the team members, identify the gaps between their knowledge levels, and recommend them reading materials that can help them to understand one another’s backgrounds so as to achieve better communication and collaboration during the project development phase.

The project development phase usually consists of several stages and various problems to solve. The project developers will seek guidance whenever they encounter new problems or receive new tasks that they are unfamiliar with. Usually, they will refer to the knowledge repository and search for learning material, technique manuals, documentation of similar projects and so on. However, this process of search is tedious, especially when the developer knows little about the field of knowledge he is searching over. It would take a lot of effort for him to find the learning materials that can provide the most relevant technical support and are most suitable for his current background, even though he is clear about the problem to be solved. A good recommender system can remarkably improve this situation. By studying the profile of the developer, the problems he is working on and his previous activities on the knowledge repository, the recommender system can provide him with the most relevant learning resources from all perspectives to help him obtain adequate knowledge rapidly.

For instances, when a user is searching for a particular learning material, the recommender system will also recommend him the prerequisite readings, the alternatives that could be more suitable for him and the team members who are expert in the field. These recommendations acquaint him with a broad range of relevant resources he has not been aware of, so that he can quickly focus on the contents that are most relevant to the problem to be solved. Similar recommendations could be given to users when they are logging into the COOPER environment, viewing the profiles of team members and tutors, exchanging messages, writing documents and so on. Through these recommendations, users are continuously informed with relevant resources; meanwhile, they can always get wanted information at hand, without spending too much time and efforts on searching over unfamiliar knowledge spaces.

Both content based recommendation and collaborative filtering based recommendation could be used in COOPER, depending on whether the intuitions behind the two approaches would apply in the considered scenarios. When a user is searching for certain learning materials, the system may recommend some relevant documents to him. Such recommendations could adopt both content based approach and collaborative approach, as the user could be interested in both the documents that are similar to what he is looking for as well as those documents that are found useful by some similar users. At the same time, the system may also recommend some experts in relevant fields for the user to communicate with. Such recommendation should adopt a content based approach, as it needs to relate the profiles of experts to learning material the user is searching for. In such case, the collaborative approach may not work very well. Many other techniques in conventional recommender systems could also be adapted to COOPER to make the recommendations more accurate and precise. These will all be found out during the next stage of COOPER development.

### 3 Latent Semantic Analysis

Latent Semantic Analysis (LSA) is usually used to assess the semantic connection between complex information objects. For examples, using LSA we can assess whether two documents are writing about the same topic. We can also estimate whether a person is an expert in a certain area by conducting LSA on the articles he has written. In the knowledge sharing framework of COOPER, LSA will be utilized to establish semantic connections between knowledge resources or between people and knowledge resources, which in turn will help to make content based recommendations and to form social networks among users based on their backgrounds and expertises. In this section, we introduce the state of the art of LSA.

Section 3.1 gives a short introduction to LSA. Section 3.2 shows several educational applications of LSA like intelligent tutoring systems, question answering and accreditation of prior learning are discussed. Section 3.3 goes into issues in text corpus construction. Section 3.4 is concerned with the strengths and weaknesses of LSA. Section 3.5 discusses the relevance of LSA to Cooper.

#### 3.1 Introduction

##### 3.1.1 Definition

Latent Semantic Analysis (LSA), also referred to as Latent Semantic Indexing, is a technique for document retrieval, or, more general, document comparison, that is based on text vector representation. Text vector representations are based on representation of a corpus of text in a matrix of terms by documents with the cells in the matrix containing frequency measures for the terms (see Table 1).

Terms	Documents																
	B1	B2	B3	B4	B5	B6	B7	B8	B9	B10	B11	B12	B13	B14	B15	B16	B17
algorithms	0	0	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0
application	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1
delay	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0
differential	0	0	0	1	0	0	0	1	0	1	1	1	1	1	1	0	0
equations	1	1	0	1	0	0	0	1	0	1	1	1	1	1	1	0	0
implementation	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0
integral	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
introduction	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0
methods	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0
nonlinear	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0
ordinary	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0
oscillation	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0
partial	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0
problem	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	1	0
systems	0	0	0	0	0	1	0	1	1	0	0	0	0	0	0	0	0
theory	0	0	1	0	0	0	0	0	0	0	1	1	0	0	0	0	1

Table 1. Example of a term-document matrix (Berry, Dumais & O'Brien, (1994, p.8))

A document is then represented as a vector of term frequencies. Figure 1. depicts a two-dimensional plot of the terms and documents from Table 1. A vector is a line from the origin through a point representing a specific term or document.

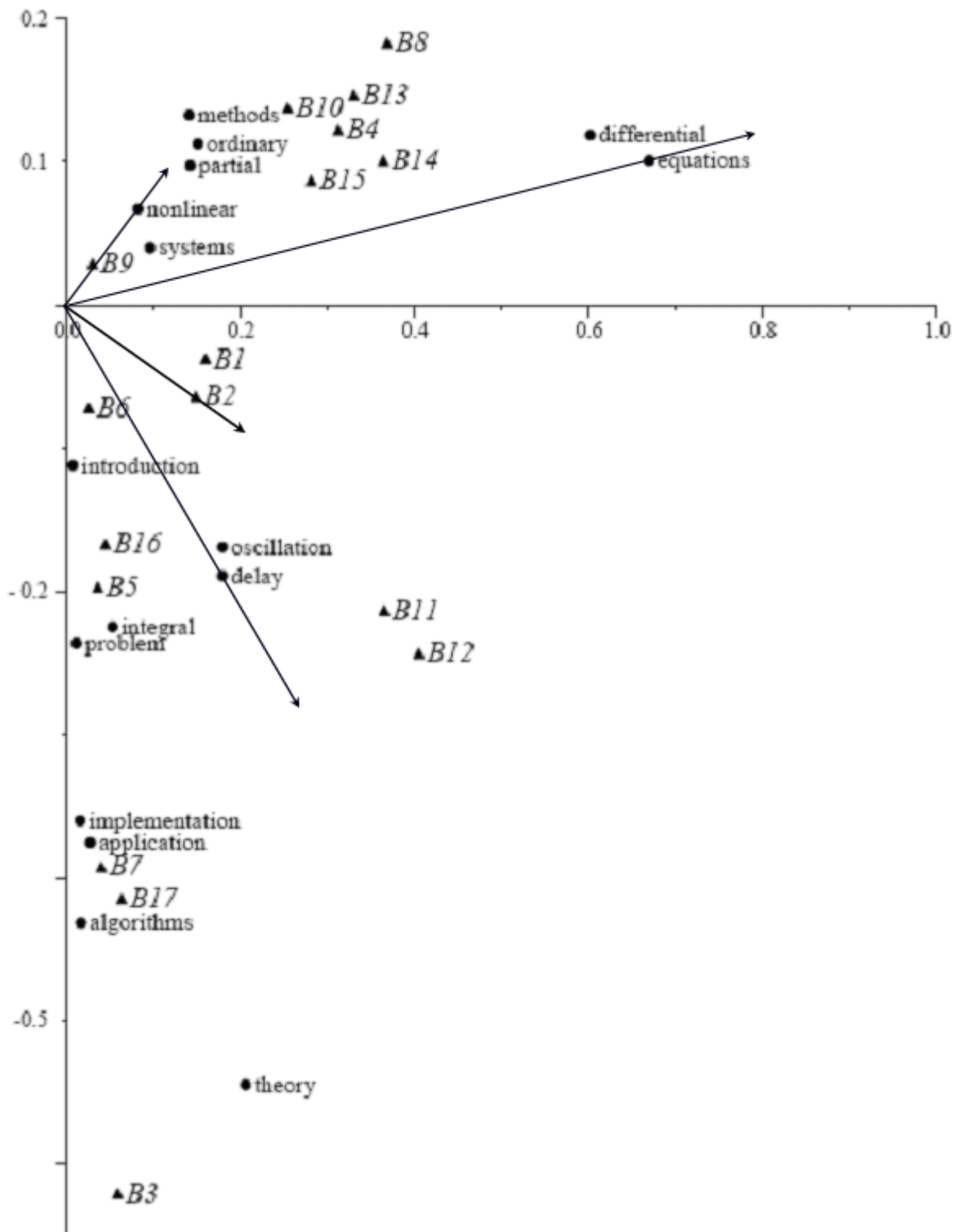


Figure 1. Two-dimensional plot of the terms and documents in Table 1. (adapted from Berry et al. (1994, p. 9))

Note that in this vector representation all syntactical information is lost (it is a 'bag of words' representation). Negation or qualifying information ("not true", "partially true") is not represented. As with any vector representation of documents, one can compute similarities between documents by computing the distance or the angle between their vectors. Latent

Semantic Analysis goes beyond these techniques in that it projects document vectors in a multidimensional space that is *abstracted* from the data.

LSA is a three steps procedure in which a data matrix is reconstructed using less dimensions than are present in the original data. In the first step, a dimensional model is obtained by performing singular value decomposition (svd) of the data matrix. This returns a diagonal matrix with the singular values and two orthogonal rotation matrices. The number of singular values greater than 0 are the number of dimensions in the data. In the second step the number of dimensions is reduced by dropping from further calculations the smallest singular values and the corresponding rows and columns in the rotation matrices. In the third step the data matrices reproduced on the basis of the reduced model (Deerwester, Dumais, Furnas, Landauer & Harshman, 1990). The underlying techniques as well as the interpretation of LSA bear close resemblance to Principal Components Analysis and Factor Analysis, both techniques that are more commonly used in social and behavioural sciences.

Semantically speaking, latent semantic analysis expresses the meaning of a text passage as a weighted sum of underlying constructs, such as contexts and concepts (Quesada, 2003). In this view, LSA is a technique to reveal these *latent semantic* variables and helps to explain the common core behind documents (context) and terms (concepts) (Landauer & Dumais, 1997). The extraction of the latent variables can be seen as a form of learning from observations. Figure 2 provides a schematic overview of how LSA is used to compare a query with a corpus.

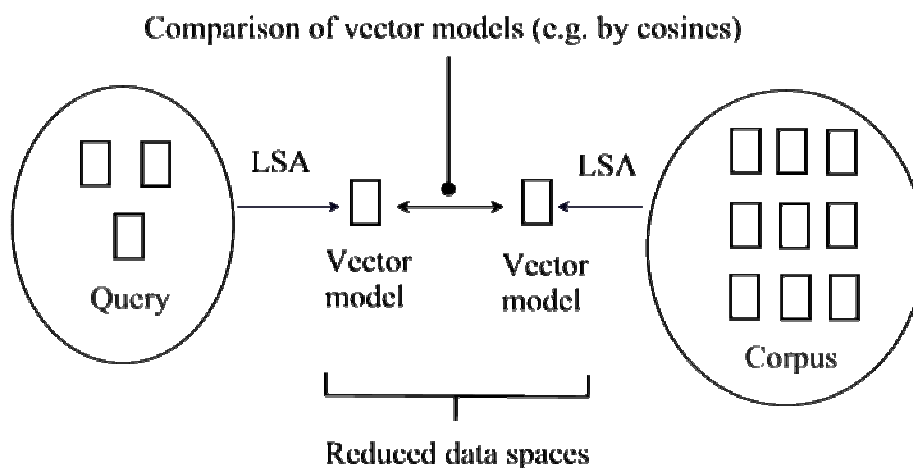


Figure 2. Schematic overview of the LSA process

### 3.1.2 LSA in a nutshell

LSA starts with a collection of terms and documents. The frequencies with which the terms occur in the documents are recorded in a table, the Term-Document matrix. A document is represented by a column vector of term frequencies (a document vector) and a term is represented by a row vector of frequencies across documents (a term vector). Note that the order of the concepts in the document is irrelevant: LSA does not log syntactic information. The dimensions of this Term-Document matrix, let's call it  $T$ , are reduced in two stages. In the first stage a singular value decomposition (SVD) of the data matrix is obtained. SVD can be seen as a generalization of principal components analysis (PCA) or factor analysis. In PCA a symmetrical matrix (e.g. a covariance matrix) is decomposed as  $C = U \Lambda U'$ , where  $U$  are the matrices with eigenvectors and  $\Lambda$  is the diagonal matrix with eigenvalues. In singular value decomposition a non-symmetrical, non-singular matrix  $T$  is decomposed as  $T = L S R'$  where  $L$  and  $R$  are orthonormal matrices and  $S$  is a diagonal matrix with singular values. The number of

singular values  $> 0$  is equal to the dimensions of the matrix. Think of the values of  $\mathbf{S}$  as defining orthogonal axes in a high-dimensional space with the values corresponding to the length of the axes. To reduce the number of dimensions, only the longest axes are retained by removing rows and columns in  $\mathbf{S}$  and the corresponding ones in  $\mathbf{L}$  and  $\mathbf{R}'$ . The original matrix is now reconstructed from these reduced matrices. In the reconstructed matrix, a document-vector may contain a frequency for a word  $W$  that did not appear in the original document. In other words, a query for "all documents about  $W$ " may return documents that do not contain the word  $W$  itself, but words that tend to co-occur with  $W$ . Several other measures can be obtained using the reconstructed Term-Document matrix, such as the correlation between document vectors. The higher the correlation, the more the documents resemble one another. That makes it possible to compare documents to each other, or compare a document to a vector of search terms.

LSA is sometimes presented as a statistical technique, but this is slightly misleading. LSA is primarily a mathematical technique. However, the core of the method, singular value decomposition, is a least squares technique that assumes, or at least performs best when the data is normally distributed and one may question whether such is the case with term frequencies (Rosario, 2000). LSA can be applied under different assumptions regarding the underlying data and this probabilistic approach may yield better results that are interpretable in a statistical way (Hofmann, 1999). Since, however, the applications of LSA reviewed have nearly all used the classical approach to LSA, we will not elaborate probabilistic approaches any further.

When we say that LSA allows comparisons of documents, the term documents should be considered in a broad sense, to cover text ranging from utterances, including search queries and sentences, to complete books. Several application areas of LSA stem from this basic approach. Thus, LSA is being used to query text databases (Berry, Dumais & O'Brien, 1994; Giles, Wo & Berry, 2001), to determine coherence within text passages or between chapters in a book (Foltz, Kintsch & Landauer, 1998), to grade essays after comparing them to one or more standards (Foltz, Laham & Landauer, 1999), to select (Wolfe et al., 1998) and sequence learning material (Zampa & Lemaire, 2002), to compare and match task and job descriptions and workers (Laham, Bennett & Landauer, 2000) or to use LSA based comparison of learning material as a basis for accreditation of prior learning (van Bruggen et al., 2004).

The application areas for latent semantic analysis can be grouped into *information retrieval* (where it is generally called latent semantic indexing), *cognitive science* and *education*. In our review we will concentrate on the first and third application areas. Uses in cognitive science, such as language learning, representation of semantics as well as problem solving and concept learning, are largely ignored in this report (see 3.2.2 for discourse processing however).

Next to the application areas, we discuss implementation issues, in particular topics around corpus construction, and some methodological considerations. Core to the latter is the determination of the numbers of dimensions to be used in the reproduction of the data.

## **3.2 Application areas of LSA**

In the past ten to fifteen years the LSA technique has been applied in a wide range of domains. In this chapter we will describe these domains more or less in chronological time order of these implementations.

### **3.2.1 Document retrieval and latent semantic indexing (LSI)**

The technique of Latent Semantic Analysis (LSA) was initially applied in the field of document retrieval (Deerwester, Dumais, Furnas, Landauer & Harshman, 1990; Dumais, 1992; Dumais, 1997; Berry, Drmac & Jessup, 1999; Letsche, 1997; Rosario, 2000; Chung-Min, Stoffel, Post, Bassu & Behrens, 2001; Freeman, Thompson & Cohen, 2000).

Deerwester et al. (1990) proposed a new approach to automatic indexing and information retrieval on the internet in the 1990's. They describe the search problem at that time as "users want to retrieve on the basis of conceptual content, and individual words provide unreliable evidence about the conceptual topic or meaning of a document. There are usually many ways to express a given concept, so the literal terms in a user's query may not match those of a relevant document. In addition, most words have multiple meanings, so terms in a user's query will literally match terms in documents that are not of interest to the user." This statement requires some explanation on how most search engines work.

Latent semantic indexing adds an important step to the document indexing process (Deerwester et al., 1990). In addition to recording which keywords a document contains, the method examines the document collection as a whole, to see which other documents contain some of those same words. It is assumed that there is some underlying, latent, semantic structure in the data that is partially obscured by the randomness of word choice with respect to retrieval. Mathematical techniques are used to estimate this latent structure, and get rid of the obscuring "noise" (e.g. common words like 'the', 'a', 'an'). Retrieval occurs by projecting the query vector (which can range from keywords to documents) on the latent structure and calculating the angle between the query vector and the document vectors, which form the latent structure. The document vectors with the smallest angles are returned. Because two documents may be semantically very close even if they do not share a particular keyword, LSI does not require an exact match to return useful results. Where a plain keyword search will fail if there is no exact match, LSI will often return relevant documents that don't contain the keyword at all (Yu et al., 2005), but have overlapping (context) vectors with the used keyword. LSA will perform better is more keywords are used, thus providing context information for the search and mapping more document vectors.

Using LSA for indexing can significantly improve three important characteristics of a search engine (Yu et al., 2005): *recall* (find every document relevant to the query), *precision* (no irrelevant documents in the result set) and ranking (most relevant results come first).

**Synonymy and polysemy** are two important issues in retrieval methods (Deerwester et al., 1990). A fundamental deficiency of many information retrieval methods is that the words searchers use often are not the same as those by which the information they seek has been indexed. Synonymy means that many different words can refer to the same concept. Deerwester et al. (1990) found that the degree of variability in descriptive term usage is much greater than is commonly suspected. For example, two people choose the same main key word for a single well-known object less than 20% of the time. Similar poor inter-rater agreement is reported in studies of inter-indexer consistency and in the generation of search terms by either expert intermediaries or less experienced searchers. The prevalence of synonyms tends to decrease the "recall" performance of retrieval systems. Polysemy, on the other hand, means that most words have more than one distinct meaning (e.g. "chip"). Thus the use of a term in a search query does not necessarily mean that a document containing or labelled by the same term is of interest. Polysemy is one factor underlying poor "precision" of search engines.

While the LSI method deals nicely with the synonymy problem, it offers a partial solution to the polysemy problem (Deerwester et al., 1990b), since the meaning of a word is determined not only by other words in the document but by other appropriate words in the query not used by the author of a particular relevant document, i.e. there is context-dependency when it comes to document retrieval. The term itself however is represented as a single term vector in the space. That is, a term with several different meanings (e.g. "bank"), is represented as a weighted average of the different meanings. Kintsch' (2001) work on 'predication' combines the relations between terms found using LSA with context-dependent information that through a spreading activation model strengthens some of the relations, while inhibiting others. Hofmann (1999) claims that 'probabilistic LSA', which is based on a latent *class* model (rather than the continuous model of SVD) and multinomial distributions of term occurrences was able to make a distinction between the different meanings of polysemic words.

The latent semantic indexing methods are capable of improving the way in which we deal with the problem of multiple terms referring to the same object (synonymy). They replace individual terms as the descriptors of documents by independent "artificial concepts" that can be specified by any one of several terms (or documents) or combinations thereof. In this way relevant documents that do not contain the terms of the query, or whose contained terms are qualified by other terms in the query or document but not both, can be properly characterized and identified. The method yields a retrieval scheme in which documents are ordered continuously by similarity to the query, so that a threshold can be set depending on the desires and resources of the user and service.

### **3.2.2 Representation of semantics and discourse processing**

Foltz (1996) and Landauer & Dumais (1997) extended the LSA technique to discourse analysis and problems with learning and language processes. They were mainly concerned with questions as: How do humans derive meaning from texts? What factors influence a reader's ability to extract and retain information from textual material? The fact that a LSA-model could 'understand' meaning of text without word order is one of the main fascinations these researchers had. They used LSA to extract and represent the contextual-usage meaning of words. The underlying idea is that the aggregate of all the word contexts in which a word does and does not appear provides a set of constraints that determines the similarity of meaning of words and sets of words to each other.

The adequacy of LSA's modelling of human knowledge has been established in a variety of ways (Landauer, 2002b). Landauer & Dumais (1997) compared human judgments of student essays to measures derived from a LSA-model, in this way providing evidence that information about the meaning of passages (semantics) may be carried by words independently of their order (syntax). They found that LSA-based measures- which take no account of word order – were as closely related to human judgments as the human judgments were to each other. They also found that LSA measures predicted external measures of the same knowledge as well or better than the human judgments'. They experimented further and found that the vectors for words derived from an encyclopaedia analysis predicted the correct answers to standardized vocabulary tests in which students are asked to judge similarity of meaning. LSA simulations matched the performance of moderately competent students. They also demonstrated that LSA 'learned' word meanings perform reading at about the same rate as late primary school children.

Foltz (1996) did something similar as the 'students essay experiment' of Landauer & Dumais (1997). He used LSA similarity measurements (in two of the three experiments by using cosines') to analyse students' essays to determine what a student learned from the original text, which texts influenced their summaries (thus predicting the source of students' knowledge) and for grading the quality of information cited in the essay. The grading was done by comparing the semantic overlap between the students essay and the original source text and between the student essay and the 10 sentences an expert grader thought were most important. A grade was assigned to each essay on the basis of the mean of the cosines between each sentence in the essay and the closest of the 10 sentences chosen by the expert grader. Next to this, in the third experiment, he also used LSA to measure the coherence and comprehensibility of texts. It is argued that the coherence of text can be calculated by examining the repetition of referents used in propositions through the text and that the degree of repetition of arguments in a text is highly predictive of the reader's recall, thus improving quality of texts. LSA predictions on the coherence of text were made by calculating the amount of semantic overlap between adjoining sentences in the text. Thus, for each text, the cosine distance was computed between the vector of sentence N and the vector for the sentence N+1. The mean of all the cosines for a text was then calculated to generate a single number representing the mean coherence for a text. They conclude that text coherence could be measured on a local (coherence between sentences) and a global level (overall coherence of the text), only that the grain size for prediction is larger for LSA than for a proposition method determining coherence of text.

Propositions represent semantic information at a clause level, while LSA is more successful in performing analyses at a sentence or paragraph level. The few words in a clause make the vectors in LSA highly dependent on the words used in that clause, whereas sentences contain enough words to permit a vector that more accurately captures the semantics of the sentence. This application makes it possible to easily detect incoherence of text and repair them, thus improving readability and learnability of texts.

Landauer & Dumais (1997) stated that the fact that LSA can capture as much of meaning as it does without using word order shows that the mere combination of words in passages constrains overall meaning very strongly. They also state that this effect depends on the dimensionality of the representation (Landauer, 2002b). Nonetheless they underscore the importance of syntax for the human meaning making processes, hypothesizing that it may reduce working memory load or ease the construction of sequential utterances.

Walter Kintsch (1998) argues that situational as well as semantic contexts influence the meaning of a concept. Concepts and their meaning expressed in propositional representations had been hand-coded until then, hence it was difficult to use them in large-scale, practical applications. He proposed LSA as an alternative to make these propositional representations, thus representing the concept in the derived LSA-vector (node in the propositional network) and the neighbouring vectors in that space as the context of a concept.

Laham (1997) also compares LSA vector space with human knowledge organization and he also argues that LSA could help to organize related concepts, by clustering concepts, which have a small difference in cosines between the LSA-vectors. This seems to work better for certain categories of words than for others (e.g. comparison of nature related words to man-made artefacts). After discovering LSA vector space as a possible representation of human knowledge organization, research was focused further on four different and more specific semantic problems: metaphor interpretation, causal inferences, similarity judgments and homonym disambiguation (words spelled and pronounced alike but different in meaning, e.g. cleave meaning "to cut" and cleave meaning "to adhere"). On the areas of metaphor interpretation, similarity judgments and homonym disambiguation's advances are made in the past years. Kintsch (2001) has developed an extended model called the 'predication model' to improve the performance of LSA compared with human performances on language comprehension. He computed the meaning of sentences with LSA, but in a context-related manner: he adjusted word vectors contextually according to their syntax in a sentence and then summed them up to compute a sentence vector. Sentence vectors of the form N1-is-N2 were computed by modifying the predicate vector N2 according to the argument vector N1. Thus, a context appropriate sense of the predicate is generated. In this way he was able to improve the LSA performance on metaphor interpretation, reaching almost similar judgments and patterns compared to humans (Kintsch & Bowles, 2002). The LSA-model also had problems with difficult metaphors, like humans, but was able to solve more easy metaphors and reaching logical solutions for the more difficult ones

The problem of causal inferences remains a recurrent problem with the use of LSA, because it does not reckon with syntactic order and relational propositions. Kanejiya, Kumar & Prasad (2003) have been suggesting an extended model as well to solve this problem, called Syntactically Enhanced LSA (SELSA). This approach generalizes LSA by considering a word along with its syntactic neighbourhood given by the part-of-speech tag of its preceding word, as a unit of knowledge representation. It also provides better discrimination of syntactic-semantic knowledge representation than LSA, but has not yet been highly successful in experimental setting. In an experiment with Auto-tutor SELSA was able to correctly evaluate a few more answers than LSA but is having less correlation with human evaluators than LSA has. Future research is needed in this area.

### 3.2.3 Educational applications

From its original inception in information retrieval, LSA has found wide application in research areas as cognitive models of human word meaning acquisition (Landauer & Dumais, 1997) and language understanding (Kintsch, 1998; Wiemer-Hastings & Zipitria, 2001), as described in the previous paragraph. Here we review applications of LSA in educational settings. Stahl (1997, in Lemaire & Dessus, 2001) suggests that LSA may be appropriate in several ways. The first concerns automatically assessing essays and providing feedback to students. The aims of the systems developed here may vary from providing (support for rating) a summative evaluation to offering formative support to students who are preparing essays or summaries. The second type of LSA educational application involves modelling the knowledge of the learner in order to select and sequence suitable instructional materials. Here, LSA is used to model both learners and instructional materials in the same multidimensional semantic space, making it possible to assess similarities between the two. The key challenge in this type of application is to select material that is in the “zone of proximal development”, thus providing the student with the right amount of new information. The third type of application is to use LSA to connect students with each other and with relevant experts, thus facilitating community formation and question answering. LSA can assess the areas of interest or the level of knowledge from the users based on their products and then suggest matches. Recently, other ideas for applications have emerged in the field, such as possible usability for accreditation of prior learning.

#### ***Intelligent tutoring systems- assessment and feedback of free text responses***

LSA has been used to assist in various assessments with various aims: helping tutors to assess students' performances, but also helping students to reach an optimal performance by providing feedback while they were practicing. Both applications are described in this paragraph. Although a difference between assessment and feedback is made, many applications can be used in both ways. The line to draw the difference between ‘assessment’ and ‘feedback’ considering these applications is very thin. Miller (2003) reviews contemporary essay-scoring systems built on LSA and mentions the Intelligent Essay Assessor, Summary Street, State the Essence, Apex and Select-a-Kibitzer in one breath. The difference made below is one mainly based on initial orientation within these projects and their (initial) main research focus in empirical experiments.

*Assessment.* LSA has been used to grade essays. Foltz (1996) compared essay scores assigned by humans to those assigned by LSA and found little difference. Foltz concluded that, at least, LSA is an automatic and fast method that permits quick measurements of the semantic similarity between pieces of textual information, thus allowing it to be used as a means to grade essays by correlating text similarity to essays of known quality. Landauer, Foltz & Laham (1998), following up on this approach, describe several approaches to automatic essay evaluation. LSA can compare the essay with defined standard(s), e.g. written by expert writers, written by previous students with a high grade, that is the so called ‘golden standard’ approach. LSA would then grade a student’s essay by applying a function on the cosines between the essay and one or more standards. Grading performance was improved by combining, with roughly equal weights, the cosine measure and a vector length measure: the former measure is sensitive of the content of the essay, the latter to the amount of content (Kintsch, 2002a). Landauer, Foltz & Laham, 1998) compared human and machine ratings using different scoring schemes such as holistic scores and topic scores. Eventually, this research led to the LSA application for automatic assessment that is probably best known in the educational community, the **Intelligent Essay Assessor (IEA)** (Foltz et al., 1999). As with any LSA application, the Intelligent Essay Assessor is trained on material drawn from the domain of the essay topic. IEA does not require a large set of graded essays. Tuning the system may require just a few examples, including a so-called “golden standard”. IEA has been found to rate essays with a reliability that matches those of human raters (Foltz, Gilliam & Kendall, 2000).

Another applications aimed at assessing student performances in essay writing is **Apex** (Assistant for Preparing Exams) (Dessus, Lemaire & Vernier, 2000; Lemaire & Dessus, 2001), developed at the Université Pierre Mendés France in Grenoble (Miller, 2003). Apex uses LSA to assess student essays on topic coverage, discourse structure and coherence. Apex differs from Intelligent Essay Assessor and Summary Street (see feedback) in that the partition of the source text in topics is much more fine grained. For calibration, the teacher must identify notions – short passages of text which exposit a certain key concept – and the topic or topics to which each notion belongs. For an essay on a given topic, Apex computes the cosine coefficient between the essay and each relevant notion and the average of these cosines' form the final score. Apex scores' were found to correlate well with human scores for content and overall essay quality. By using the notions, Apex is able to construct an outline view of an essay, thus helping students in planning the discourse and also highlighting areas of concern. The outline is produced by having LSA find and print each essay paragraph's closest corresponding notion. If no notion correlates above a certain threshold, the paragraph is flagged as potentially irrelevant. The completed outline also helps to identify repetitious sections. Apex also performs coherence analysis (comparative to (Foltz et al., 1998) by comparing adjacent sentence pairs and reporting abrupt topic shifts).

*Feedback.* Several projects attempted to use LSA to provide (faster) feedback to support self study. The feedback allows students to engage in extensive independent practice without placing excessive demands on teachers for feedback. Detailed feedback often requires that LSA operates on more fine-grained aspects of texts. For instance, the coherence of a text has been measured by calculating cosine similarities between individual sentences. A high overall similarity indicates repetition or rephrasing of the text, while an overall low similarity is an indication that the text has a low coherence. Drops in similarities between successive sentences can indicate topic breaks. A high average number of topic breaks may indicate that a text jumps from topic to topic.

These types of measures are used in **Select-a-Kibitzer** : (Wiemer-Hastings, Wiemer-Hastings & Graesser, 1999; Wiemer-Hastings & Graesser, 2000), an educational software system that provides feedback on student compositions. Once a student has entered her text, specialist agents - the Kibitzers - may be invoked to provide feedback on the particular text characteristics in which each of them specializes. Each kibitzer acts as a critic for a particular discourse feature, be it stylistic, grammatical or semantic. LSA is used to determine the coherence of the text, and the topic breaks between the sentences are used to identify semantic chunks. The sentence with the highest average similarity to other sentences in the chunk is considered the key sentence and is presented as the system's understanding of the topic. Like Apex, Select-a-Kibitzer generates outlines of essays, but it does so without reference to a source text. The program uses clustering methods on the LSA semantic space (like Laham, 1997) to identify discrete topical chunks in the corpus. For each chunk, the program selects as an archetypical sentence the one that compares best with all other sentences in the chunk. An outline of key points is then produced by printing the selected sentences in order of appearance in the essay. This outline gives the student an idea of the essay's progression of ideas, something particularly useful for beginning writers. The LSA engine in Select-a-Kibitzer is also trained in template sentences to help determine the purpose of sentences. Templates such as "I would change....because" are used to indicate why-reasoning.

**State the Essence** (Kintsch et al., 2000) was designed to improve elementary schools student's summarization skills, helping them mediate the conflict between concision and comprehensiveness (Miller, 2003). After an initial spell-check, LSA is used to measure topic coverage, irrelevancy and redundancy. The system does not provide feedback on other aspects of writing, such as sentence structure, organization and style. Because of LSA's ignorance of syntax or morphology (Miller, 2003), it cannot judge most matters of mechanics and style (e.g. spelling, grammar, clichés, tense shifts). Students can revise and resubmit as often as they like. Once they are satisfied with the feedback, they submit their papers to the teacher for complete grading. Initial trials of State the Essence indicated areas for improvement. Overall correlation

with humans was inconsistent and there was no evidence that use of the program resulted in increased writing skills or learning. More seriously, students tended to forget or ignore the fact that the program was evaluating content only, preoccupation with the numerical score incited many students to abandon good writing style in favour of increasing their score by the cheapest means possible. They received heavy penalties for organization and mechanics upon human grading.

Hypothesizing that the bulk of the problem lay in the feedback mechanism, Kintsch et al. (2000) revised the system. Visualizations of numeric scores and changes on how and when advice on redundancy is presented were made. This new version was renamed to Summary Street.

**Summary Street** provides various kinds of immediate feedback, primarily about whether a student summary adequately covers important source content (based on several 'golden summaries' and representations of the source text) and fulfils other requirements, such as length. It tells students what information in the source is missing, provides comments on redundancy and relevance.

Experiments with Summary Street suggest that it is especially helpful when students are faced with more difficult tasks or with a harder text. Kintsch (2002b) and Wade-Stein & Kintsch (2003) reported three notable results of using system feedback while writing summaries. Time on task increased significantly when students could use the system: students were willing to work harder and longer when given immediate feedback. Summaries written with content feedback received higher grades from the teachers. This was the case for difficult summaries, for which grades more than doubled, whereas for texts that were easy to summarize, the use of the system had no significant effect. The researchers also observed a transfer effect. Students who had written a summary in the previous week with the help of *Summary Street* wrote better summaries even when they no longer had access to the feedback the system provides.

Magliano, Wiemer-Hastings, Millis, Munoz & McNamara (2002) tested a computer-based procedure for assessing reader strategies that was based on latent semantic analysis (LSA). During a computerized version of self-explanation-reading-training (SERT), students read texts and typed self-explanations. Self-explanation refers to explaining difficult text to oneself. The strategies include using logic or world knowledge to elaborate on the current sentence (knowledge building explanation), making conceptual bridges among ideas in text and predicting what will come next in the text (sentence-focused explanation). A minimalist approach would be to paraphrase the sentence or provide a vague description. The goal was to test if LSA could be used to assess the extent to which students used these strategies and classify the self-explanations as 'knowledge building', 'sentence focused' or 'minimalist'. Several semantic benchmarks were used as a reference model for students' self-explanations: (1) the current sentence (2) causally important prior sentences (3) relevant world knowledge and sources that the reader can draw upon while self-explaining. For the last, a semantic space on heart diseases created by researchers at the University of Colorado at Boulder was used (also see <http://lsa.colorado.edu>). The hypotheses were that knowledge-building self-explanations should have a high overlap (e.g. high cosines) with causally important information from the prior text and/or relevant world knowledge. In contrast, a minimalist self-explanation should have a relatively low overlap with the prior text and relevant world knowledge, but a relatively high overlap with the current sentence, because the reader is primarily paraphrasing the sentence. A sentence-focused self explanation should also have a relative high overlap with the current sentence, but should have intermediate overlap with the prior text and relevant world knowledge. The LSA-assessment was compared with human judgments of the self-explanations. Both human judgments and LSA were remarkably similar and indicated that students who were not complying with SERT tended to paraphrase the text sentences, whereas students who were complying with SERT tended to explain the sentences in terms of what they knew about the world and of information provided in the prior text context.

### ***Intelligent tutoring systems- selection and sequencing of instruction***

LSA has been used to select instructional text that is appropriate to the student's background knowledge, i.e. a text that matches the prior knowledge of a student partly, but also adds new concepts to it. Appropriate text is neither too easy, nor too hard for a student.

Rehder et al. (1998), Wolfe et al. (1998) and Landauer (2002b) began to use LSA to match students with text at the optimal level of conceptual complexity for learning. LSA was used to characterize both knowledge of an individual student before and after reading a particular text and the knowledge conveyed by that text. Wolfe et al. (1998) addressed a similar issue, referring to it as the "zone-of-learnability". The key to their approach was to select a study text to match the prior knowledge of the learner as closely as possible. First, they collected data on the students' prior knowledge, and then had the students study one of four different texts about a topic such as the anatomy, function and purpose of the human heart and the circulatory system. The texts ranged in difficulty from elementary school to medical school level. As expected, learning gains were related to prior knowledge: texts that were too easy or too complex yielded weaker learning gains. Wolfe et al. (1998) presented a number of curve-fitting solutions that relate LSA-based similarity measures between prior knowledge and the study texts to predict learning effects. If the cosine between an essay written by a student and an instructional text was moderate, learning was successful (around 40% improvement in test scores or essay grades between pre- and post-test); when the cosine was too low (not enough background knowledge) learning was poor, when the cosine was too high (not enough new information in the text), learning was equally poor. Zampa & Lemaire (2002) used LSA in an intelligent tutoring system to model a domain and the student and select appropriate texts to match students knowledge level. In their model, a domain is built of "lexemes", being either words in a language-learning domain, or facts and conclusions in a problem-solving domain. Note that this domain representation is not based on raw text, but requires prior identification of the lexemes. The student, it is assumed, learns the domain by being exposed to a series of lexemes. The tutoring system selects those texts/topics in a zone around the student and domain sequences that have already been addressed. Sequences that are too close or too remote are expected to yield a weaker learning effect and are therefore ignored.

AutoTutor (Wiemer-Hastings, 1999; Wiemer-Hastings et al., 1999) engages students in a natural language conversation and thus encourages them to provide elaborate answers to the questions it poses. AutoTutor scores the quality of the answers that the students provide in conversational turns using a variety of techniques, including LSA. AutoTutor rates the quality of the students' assertions much the same as intermediate-level experts, but not as well as accomplished experts. The LSA component of AutoTutor is able to discriminate between classes of simulated students, and is capable of tracking the increased coverage of a topic in successive turns. Another purpose of Autotutor is to answer unrestricted student questions (Lemaire & Dessus, 2001). It does so by selecting the closest piece of text to the question. Recently (Graesser et al., 2005), implementations of AutoTutor in different domains have been made and LSA has generally been successful in evaluating the quality of student explanations and assertions in tutorial dialog.

### ***Community formation and community support***

(Stahl, 1997) already mentioned possible usability of LSA to connect students with each other and with knowledge experts. (Yukawa, Kasahara, Kato & Kita, 2001) have implemented this idea in an expert recommendation system. This system processes the description of a technical topic as input and then find engineers who have a high level of expertise in that area. Also relevant documents are retrieved. The technique is an extended vector space model that locates both technical topics and engineers in the same multi-dimensional space and then calculated their relevance. This system can also retrieve engineers or documents that are related to a field matching a given engineer's technical interests.

Recently this idea is elaborated upon for transient communities: communities that fulfil a specific goal and exist for a limited amount of time. Interest as well as expertise areas of members could be based on comparison of the document sets that members of the learning communities collect and produce (Kester et al., submitted). In this way, ad hoc, transient communities could be formed, based on questions asked by a community member and answered by an ad hoc formed group of peers, who have knowledge on the topic and are stimulated by 'seeds' (little fragments of contents which seem to be relevant and are selected with the help of LSA). Kester et al. (2005) proposed a model for this and are planning to experiment with an implemented version of this model within the Agents for Support Activities (ASA) project (Croock et al., 2003) at the Open University of the Netherlands.

### ***Question answering***

Within the same ASA-project one model for an agent for support activities is based on question-answering (Croock et al. 2003). The basic idea is that students pose natural language questions to a database and the database will provide the most relevant (part of a) document to provide an answer. This is not a new idea. Caron (2000) reports on a prototype system for technical support called the Frequently Asked Question Organizer (FAQO). This application enables technical support personnel to construct a knowledge base from email archives and other existing documents. Users can query the knowledge base using natural-language questions in order to find relevant documents. The prototype that used LSA for query matching outperformed the keyword-search tool that was previously used. In a recent experiment, community information is used as an additional source of information to specify context for a certain question (Almeida & Almeida, 2004). The community-based information was used in order to provide context for queries and influenced by recent interactions of the user with the service. The algorithm used was tested on the service of an online bookstore. The quality of content-based ranking strategies in this way could be improved significantly and retrieval was improved with 48%.

### ***Accreditation of Prior Learning and positioning***

Van Bruggen et al. (2004) and Koper, van Bruggen, Rusman & Giesbers (2005) propose to use LSA to position learners in learning networks. Because learners can enter and leave such a network as they like, there is a recurrent need to position them in the right place within the network. In order to prevent the learner from taking redundant or too complex learning material and to accredit prior learning, latent semantic analysis is proposed as a tool for learner positioning in learning networks (van Bruggen et al., 2004). The core assumption is that equivalence of outcomes will be reflected in, or can be approximated by, the similarity of the contents of (learning) materials studied or produced by the student (source material) and the material contained in the learning activities in the learning network (target). LSA is used to compare the contents of a learner's portfolio with the contents of learning materials contained in learning activities.

### **3.2.4 Human Resource Management**

Another domain in which LSA is applied is Human Resource Management. A prototypical tool (HEADHUNTER) that matches jobs, people and instruction is worth mentioning here. Laham et al. (2000) experimented with LSA to match jobs, people and instruction in an air-force setting. Their aim was to help identify required job knowledge, to determine which members of the workforce had required job knowledge, pinpoint needed content which could be (re-)used within training settings and to maximize training and retraining efficiency. They processed data on three Air Force occupations for which full job descriptions were available. They then analysed "duty lists", tasks grouped into functional units, and individual tasks along with the tasks, which were actually completed in practice, thereby constructing a single semantic space for jobs and people. The semantic similarities between jobs and people could be used to decide between candidates for the job or to select a replacement.

It appeared that LSA could help to characterize tasks, occupations and personnel and measure the overlap in content between instructional courses covering the full range of tasks performed in many different occupations, thus indicating where the wheel was invented twice in the same working and training setting. It showed that it could estimate the similarity of each task or occupation to every other task or occupation, measure the degree of match of each airman to every task of occupation, estimate which airmen could most easily take the place of others and indicated that LSA has the potential to identify in detail possible re-usable knowledge components and match the knowledge components required by new systems with those contained in segments of existing training materials and with the experience of individual airmen. The experiment was based on a database with 20000 documents. The importance of LSA with large databases (e.g. for thousands of personnel in large branches (e.g. military or international corporations) was emphasized. LSA allows analysis that have been heretofore impossible because of the size and complexity of the data involved. Laham et al. (2000) also suggest that the system could also be trained to predict which course would bring a person closer to a target job profile.

Two experiments with a later version of this agent software, called CareerMap, again in an Air Force setting are reported in Laham, Bennett & Derr (2002). In the first experiment, LSA is used to analyse course content and materials that are used in the current training settings and to identify appropriate places in alternative training structures where that content can be reused. This saves time for training developers since the pre-existing content has already been validated as a part of its earlier application. Also gaps in the content for the new training structure become readily apparent. The second experiment is an implementation of a combined speech-to-texts (verbal communications translated to text) and LSA-based intelligent software agent for embedding automatic, continuous and cumulative analysis of verbal interactions in individual and team operational environments. Currently it is impossible to evaluate verbal communication to identify critical information and content required to operators. LSA has potential for assisting operators in the performance of their tasks because it can 'listen' and in almost real-time evaluate free-form communication from a variety of sources and match content to stored language dictionaries. One application of this technology being explored is tracking and scoring the tactical communications to identify areas of training need and as an additional tool for assessing the efficacy of scenarios and missions. Both experiments reported positive results with the use of LSA.

### ***3.3 Implementation issues: corpus construction***

LSA requires a text corpus and this chapter discusses several topics to consider when creating a corpus, such as corpus size, document size and document selection and related issues such as filtering and tidying, including stemming and stopping.

We do not discuss attempts to include additional, semantic information in corpora. Some semantic pre-processing of the corpus by identifying lexemes (words) (Zampa et al. 2002), by segmenting and breaking down corpus elements (sentences) by hand (Wiemer-Hastings, 1999) or by using an (automated corpus training) method of speech-tagging (Wiemer-Hastings & Zipitria, 2001) is reported in the literature. The LSA performance of both did not match human judgments as close as standard LSA did. This type of pre-processing is therefore omitted in our further discussions.

#### **3.3.1 Corpus size**

Most discussions of corpus construction are concentrated on the size of the corpus. However, it is not always clear what is meant by "large" and "small". The same is true for what size is perceived as a minimum and/or maximum requirement to successfully apply LSA.

LSA is often used for document retrieval from very large document databases, containing ten thousands of documents and an input of 5000 documents to train LSA on the domain is quite common in these applications. Deerwester et al. (1990) state that a "reasonable size" is 1000 to

2000 abstracts which means about 5000-7000 index terms. In contrast, in Laham et al. (2000) a total number of 20.000 objects is mentioned as a very small dataset compared to LSA's capabilities, but enough to do a fair job in estimating statistical regularities. Many authors seem to take this as a minimum requirement.

Several researchers show that big corpora are better, *but*, according to Landauer et al. (1998) the *goal* of using LSA is an important factor. They would like to "*truly represent the sum of an adult's language exposure*" (p.35) of which they state that it is impossible because (1) it's impossible to gather such a big corpus and (2) current computational power is not enough to perform SVD on 100.000's x 10.000.000's matrices. Educational uses of LSA, in contrast, are often confined to smaller corpora, that are more specific to (sub)domains. Within these corpora LSA has been shown to be robust against decreasing the size of the corpus (Wiemer-Hastings et al. 2000). According to Wiemer-Hastings et al. (2000), the best corpus is specific enough to allow for subtle semantic distinctions within a domain, but is general enough to ensure moderate variations in terminology won't be lost. They report a 'graceful degradation' of performance. When they reduced the size of their text corpus from 2.3 MB to a minimum of 15 %, the performance of LSA in terms of correspondence with human raters decreased 12%. These results are clear indications that LSA can perform reasonably well in small scale corpora. More empirical research is desirable, especially because: "*A smaller corpus takes less time to train, less storage space, and less processing time for comparisons. Thus, if there is no significant performance advantage with larger corpora, they can be avoided*" (Wiemer-Hastings & Graesser, 2000, p.7).

Further research confirms that it is possible to obtain meaningful LSA results from smaller corpora. For example, Wild et al. (2005) used 43 files each consisting of a students' answer on a marketing question from a real world exam. They performed many test runs (2016 in total) to see what the influence of different parameters like pre-processing on the correlation between machine scores and human scores would be. Results show significant correlations between these scores, which means that LSA can work well on small corpora. Our own experience shows that meaningful results can be obtained by using 287 documents (about 10000 terms), each consisting of a single paragraph of text on monkeys and/or apes.

### 3.3.2 Document selection

Document selection mainly concerns the question whether it is better to have a large collection of *general* conceptual content than a small collection of more *specific* conceptual content.

The number of documents may not be the main issue for our purpose of using LSA because the corpora used in learning networks are specific to particular sub-domains. It is obvious that in these corpora the dimensionality in the data is less than in broad corpora such as those build on the basis of a complete encyclopaedia. The number of documents needed for LSA is not dependent on the size of the domain (or text database) but on the dimensionality of the domain.

Like corpus size, the "ideal" number of dimensions also is ambiguous. Finding the correct number of dimensions is critical because if it is too small the structure of the data is not captured. If it is too large the latent structure cannot emerge and all unimportant details and sampling error remain. In general, the "magic" number of dimensions was reported between 100 (e.g. Deerwester, Dumais, Furnas, Landauer & Harshman, 1990; Wolfe et al., 1998; Wiemer-Hastings et al., 1999) and 300 (e.g. (Landauer, 2002a; Franceschetti et al., 2001; Olde, Franceschetti, Karnavat, Graeser A.C. & Tutoring Research Group, 2002).

When working with small corpora (< 300 documents), the number of dimensions will obviously be smaller. Recent research showed that it is very well possible to make meaningful use of LSA in small corpora of which the "ideal" dimensionality can be about 40 (Nakov, Valchanova & Angelova, 2003). Below we will discuss some of our own research that yielded similar results. After gathering a collection of documents that grasp the dimensionality of the domain in the best possible way, there are a number of things to do to tidy the corpus. This means that

elements that are meaningless to LSA like html code, diacritic tokens and images are removed. Furthermore, it may be desirable to remove redundant text, identical text and spelling errors. Other more specific pre-processing techniques like stopping and stemming are discussed in a separate paragraph.

### **3.3.3 Document size**

Documents for LSA may be as small as individual sentences or as large as essays, articles or web pages. Research reports mention a variety of documents, such as student answers on a test question (e.g. Wild, Stahl, Stermsek & Neumann, 2005), encyclopaedia articles (e.g. Wolfe et al., 1998), parts of textbooks on a certain topic (e.g. Wiemer-Hastings et al., 1999) or student essays (e.g., Rehder et al., 1998). Reports on document size or grain size vary from an average document size of 50 words (e.g. Deerwester et al., 1990) to full articles on a tutoring topic (e.g. Olde et al., 2002). Often full text articles are cut into smaller parts, which are about one paragraph in length. Wiemer-Hastings et al. (1999) state that the paragraph is said to be, in general, a good level of granularity for LSA analysis because a paragraph tends to hold a well-developed coherent idea (Peter Foltz, personal communication, October 1997). This is supported by findings from (Rehder et al., 1998) who found a minimum essay length of 60 words suitable for the purpose of knowledge assessment.

In our own experiment each document was manually split and most of the time matched one paragraph containing information on a single species of monkey or ape.

### **3.3.4 Stemming**

Filtering the data of a corpus can be done by stemming and by stopping. Stemming refers to the parsing of tokens to their semantic stem. Thus, tokens such as "hypothesis", "hypotheses" and "hypothesized" would all be stemmed to a semantic root "hypothes". Stemming has the potential of raising the semantic relevance of the results. For example, the "mountain gorilla" is called the "bergländgorilla" in Dutch, which is parsed as a single, completely different token than "gorilla". In addition to stemming the corpus, stemming a query can also raise semantic relevance. With respect to LSA, stemming alone is reported to show only one to five percent improvement (Dumais, 1992) or even to reduce average correlations with human raters (Wild et al., 2005).

### **3.3.5 Stopping**

Stopping refers to the filtering of noise words, such as "the" and "not" that occur frequently and indiscriminately in the corpus. Noise terms appear throughout any "raw" corpus and do not contribute to the discrimination of documents. From a measurement point of view these terms only add error variance to the corpus. In large corpora that reflect broad domains, such as those based on an encyclopedia or a collection of web sites, the terms to be stopped may be determined by consulting a general list of word frequencies in the written language Wild et al. (2005) indicate that stopping is absolutely necessary. Our results support this conclusion.

## **3.4 Evaluation**

In order to determine why LSA should or should not be used, an evaluation is provided which reflects the strengths and weaknesses of the technique.

### **3.4.1 Strengths of LSA**

The main advantage of LSA is that it allows for fairly intelligent operations to be performed by putting in a minimum amount of effort. This is illustrated by its ability to use word co-occurrence data to move words and documents into a reduced dimensionality space where they can be more meaningfully compared to each other. This is fully automatic and does not require the use of metadating, preliminary construed dictionaries, semantic networks, knowledge bases,

grammatical syntactic analysers etc.. Automation of processes can alleviate human workload considerably which is an additional advantage.

### ***Strengths through mathematical representation***

With respect to other mathematical techniques it has been said by Miller (2003) that LSA concerns inter-word relationships at a deeper level than co-occurrence measures ever could.

Besides saving time, it has been found that the quality of its output can be very high. Essay grading systems which use LSA, consistently outperform those without. For example, over diverse topics, the Intelligent Essay Assessor scores agreed with human experts as accurately as expert scores agreed with each other (Foltz et al., 1999). LSA predicts scores as well as human graders (Landauer & Dumais, 1997); (Wild et al., 2005) and LSA can measure prior knowledge well enough to select appropriate text (Wolfe et al., 1998).

Because LSA uses a mathematical representation of the relations between words in a text and the semantic distance between texts it offers a rapid analysis of large numbers of documents.

As described in section 3.2.3, LSA has several applications in education. Because of its possibilities to use with respect to dynamic corpora (also see chapter 5) LSA seems ideal for use in learning networks.

### ***Strengths through representation and reduction of complexity of concepts***

If the internal representation of semantic similarity is not reduced but constructed in as many dimensions as there are contexts, there would be little practical use for the output of LSA. The strength of LSA is that it represents a corpus in a  $k$  dimensional space and thereby reducing the complexity, which makes it possible to improve the estimates of pair wise similarities. It is hereby possible to accurately estimate the similarities among pairs never observed together, by fitting them as best we could into a space of the same dimensionality. For example, research done by Deerwester et al. (1990) shows an LSA model of relations between 60,000 words (30,000 text passages) made with LSA to score on a synonym test for admission in U.S. College to perform as well as the average student who did the test.

### **3.4.2 Weaknesses of LSA**

The weakness of LSA lies in the empirical determination of computational factors, the computational time that is needed to analyse big corpora, the directionality of LSA and the application in contexts with the emphasis on logic and reasoning.

#### ***Empirical determination of computational factors (e.g. singular values)***

The number of singular values that offers the best result is no fixed “magic number”. It is very important to determine the right number of dimensions for the amount of success of LSA (Landauer, 2002b).

Often operational criteria are used as a way to determine the ideal number of dimensions, that is, the number of dimensions which delivers the best result is determined (it is probably highly dependent on what kind of result is aimed to be acquired). As seen in chapter 3.3.2, the generally accepted “ideal” number of SV’s lies between 100 and 300 which provides no option if we work with small corpora. We suggest an additional rule of thumb for determining suitable singular values within small-specific corpora: the explained amount of variance.

#### ***Computational time for dynamic corpora***

As mentioned in section 3.3.1, (Landauer, Laham & Foltz, 1998) state that current computational power is not enough to perform SVD on 100.000’s x 10.000.000’s matrices. (Quesada, Kintsch & Gomez, 2001) also mention the demand for powerful computers to

perform necessary analyses. For our goals in Cooper, large matrices like Landauer et al. and Quesada et al. use may not be needed but computation time may be a problem because of the dynamic nature of material we want to use. The gravity of this problem is determined by the frequency with which the matrix is updated: if this is once a week or even once in a few days the problem is decreased significantly.

#### *'Directionality' of knowledge*

All LSA input, like an essay for example, is represented by a vector. The direction a vector has, is interpreted as the representation of the quality of the semantic content of that particular piece of input. The cosine does not provide any information about the "directionality" of knowledge because it measures relatedness as an unsigned angle in a high-dimensional space (Rehder et al., 1998). This means that the essays of two students may have the same cosine in comparison to an instructional text, but one of the essays may be dissimilar to the text because the individual knows very little about the topic (relative to the text), whereas the second essay may be dissimilar to the text because the individual knows very much about the topic (relative to the text). This can be solved by using a combination of LSA and multi-dimensional scaling (MDS) (Rehder et al. 1998; Carol & Arable, in press). MDS is used to (re)calculate subspaces and can compare the distance between different input texts for example to make a distinction between novices and experts. Three methods are available of which one is recommended as most accurate. This method calculates cosines between all pairs of text are to create a detailed Euclidean space. Then, a MDS procedure takes place using a standard procedure (Carol & Arable, in press) to create a 10-dimensional space that represents all non-random differences between the cosinusses. This method is effective, but also contains a step of empirical "matching" of the parameters and therefore it is more sensitive to chance and variability.

#### **Reasoning and logics**

All LSA models are based on co-occurrence of concepts in documents. The order in which these concepts occur/co-occur is completely ignored. This means that LSA is inadequate to detect logical fallacies. As Wolfe & Goldman (2003) point out, LSA fails to represent domains in which the context determines how sentences should be interpreted. This applies to domains that use metaphorical language, causal reasoning and logically ordered sequences of steps.

Fooling an LSA based essay grader by submitting an essay with just keywords only is possible but will not be a problem. Of course, results will be contaminated, but if a student is capable of writing such an essay, s/he has a very good understanding of the domain and will have proven so (Lemaire & Dessus, 2001).

### **3.5 Relevance of LSA for Cooper**

#### **3.5.1 Community formation, community support and collaboration**

LSA can play a role in community forming as well in the support of communities while they are collaborating. Interest as well as expertise areas of members can be based on comparison of document sets that members of the learning communities collect and produce, thus providing shared interests and probably shared goals/aims between community members from the start of the group formation. This 'social matching' process could be supported by the provision of a visualization (as a part of an identity representation) based on the topics people are interested in (comparable with Flickr's visualization of its folksonomy, <http://www.flickr.com/photos/tags/>). This could help to form informal (sub)groups within the large community of learners, tutors, alumni etc.

Also, while people are collaborating within a community, certain questions will rise. To find the people who are experts on these topics and will probably be able to answer the question, LSA could be useful in the matching of questions and interest areas. Small temporarily sub-groups within the community could be formed, like Kester et al. (2005) do with their transient

communities, to solve the 'problem' (=question) based on background and expertise of members.

In both examples of LSA's usability for communities it becomes easier to find information, like who is doing what and what topics are related. It also helps to match members interests and aims, which can increase motivation to participate. It mainly focuses on matching people to people.

LSA could also help to create a feeling of trust within a group of collaborating people with a specific task and goal within a specific time span, but who don't know each other and don't have the opportunity to see each other (a lot). LSA could help to make a certain representation of e.g. interest areas of these people, thus helping other project members to form a mental image of the other person, without spending a lot of time on this image building. This first image is important for the forming of trust, which ultimately has an influence on development of group conflicts and on group performance and interactivity as a whole.

### **3.5.2 Human Resource Management and task allocation**

Other matches could be made with the use of LSA: to position people on the 'right' job by matching descriptions of people to a job/role profile, to provide people with the 'right' instruction or to provide people with the 'right' mentor ('right' meaning as personalised to the need/question as possible). Based on a question or the specified need of a student, suggested instructional material could be filtered or a mentor selected based on his/her profile with expertise and interests. In this relatively new field of research and practice, thus far interesting and promising results were obtained within the domain of the army (e.g. (Laham et al., 2000), where functional matches between jobs, training and people were made.

### **3.5.3 Localizing resources**

The descriptions above are quite specific applications of LSA. But in both cases, a specific resource is located for a specific user and aim/need/question. In general, LSA is very useful to localize resources: it can compare and determine similarity between two text-based sources, thus determining compatibility. In this way it could be useful in many ways, e.g. localizing experts (within and out of a certain group), localizing documents (e.g. of previous project groups/comparable projects), localizing group of interest and localizing potential sponsors.

### **3.5.4 Support of assessment and feedback**

When project deliverables are largely comparable to previous projects, LSA could also play a role in the final assessment process of projects (like the assessment of airplane landing technique in Quesada (2003). It could support assessors in their judgement, e.g. to compare the deliverable to previous high quality project deliverables, which were qua problem, domain and content more or less the same. In this way, it could provide a type of framework for the judgement of deliverables.

For students, it could help them to consider alternative perspectives on a topic, by providing feedback and suggestions on related topics while they are working. E.g. suggestions like "previous project groups also considered/mentioned 'x' and 'y' while they were working on this topic".

## 4 Knowledge Sharing Systems

Knowledge related activities in the COOPER context could be viewed through different lenses, namely “know what”, “know who” and “know how” [19]. “Know what” deals with knowledge repository, knowledge management, information structure, etc. “Know who” deals with knowledge practitioners, experts and their network, and “Know how” deals with the activities of networking, sharing, collaborating and so on. The knowledge sharing systems referred by this section mainly concerns the “know who” and “know how” activities. In other words, they are systems that can help to gather people together and provide them tools to share and construct knowledge.

Different from ordinary information exchanged over the internet, which can be understood individually and in isolation, knowledge can be understood only within a context, through interactivity and communication [Konstan, J. A. et. All, 1997]. As it is said, “Knowledge can be regarded as the only unique resource that grows when shared, transferred, and managed skilfully” [Alfred Beerly]. Collaboration and sharing are essential for the progress of knowledge growth. In today’s computer-aided knowledge sharing systems, knowledge becomes available as digital content, which is produced by individuals that are part of intersecting networks of interest and communities of practice. A simple manner to create knowledge is by asking a question and watching as the answers create cascading conversations and interactions among the network users. Through such networks knowledge is refined, reinvented and reinterpreted in an automatic manner. The best example for this type of knowledge sharing and refining is the online encyclopaedia, Wikipedia.

In this section, we give an overview of the most widely used knowledge sharing system on the internet, and discuss how they can be used to improve the knowledge sharing and knowledge co-construction in COOPER.

### 4.1 Knowledge Sharing Applications on the Web

The internet and the Web have been providing a world of opportunities for collaborative knowledge construction. Many tools like e-mail, discussion forums, chat are already popular in educational environments. Recent innovations like blogs, wikis and RSS feeds offer powerful opportunities for online collaboration and knowledge sharing. We elaborate on these applications individually.

#### 4.1.1 Discussion Forums

Discussion forums are also known as internet forums, web forums, message boards, forums and so on. As an extension of emails, discussion forums facilitate group exchanges of information. The rough definition of a forum is the ability for people to start threads and reply to other people’s threads. They automatically maintain a log of all messages in a threaded, hierarchical structure. A major difference between forums and electronic mailing lists is that mailing lists deliver emails to the subscriber (pushed), while for reading forums the user has to visit the website and check for posts. Forums can also offer email notification in the case of new entries. In a forum regular users are not allowed to edit other people’s entries, they can only post comments. Discussion forums are seen as tools which encourage universal participations to discussions compared to the face-to-face dialogue in emails. Virtual communities develop quite often around forums with regular users.

Discussion forums are well used on the web, and have helped to establish uncountable virtual communities among people with different cultures, nationalities, interests, professions, etc. Some well known forums include GameFAQs , Gaia Online, and IGN.

### 4.1.2 Blogs and RSS feeds

Blogs (also known as web logs, or weblogs) are essentially highly interactive online journals. Blogs are web sites that contain frequently updated posts that are usually displayed in reverse-chronological order. Writers can use hypertext to link to what others have written on given topics or to external resources. Blogs permit other users to post comments which are logged and become visible from within the blog page. Usually blogs are created by individuals, but there are also group blogs that contain highly interconnected blogs which form communities. Technology-related blogs are group blogs that form a large, loosely interwoven net of information, in which blog entries are linked and debated. There are also topic related blogs, one of the most visited blogs is <http://www.instupundit.com>.

Blogs are about sharing information, ideas and resources. When they first appeared, blogs were not so oriented towards a collaborative environment [Brady M. 2005]. There have been three major additions to the blog: permlinks (permanent links), comments and trackback. Permanent links point to the individual blog posts, users have the possibility to place comments to other user's posts and trackback is a citation notification system. When many blogs have common topics they will eventually link each other and discussions will start. One of the criticisms of blogs is that they consist of personal opinions of individuals who usually are not experts on the discussed topics.

Most blogs are personal or journalistic. In education, blogs are used as personal journals for students, where they can link the blogs to different courses and use them as personal electronic portfolios, tracking the development over time. Blogs can be used as environments for project-based learning, however a limitation of the structure is that they are chronological organized, rather than by content. Indices and search mechanisms on blogs can also be used for finding information.

*RSS feeds* are used to alert users about new blog postings and also to help sort information coming from different blogs and other Internet resources. RSS stands for "really simple syndication" and was first developed by Netscape as a way for users to add "channels" to MyNetscape pages. RSS is a family of web feed formats, specified in XML and used for Web syndication. They are mainly used by news websites, blogs and podcasting. Web feeds provide web content or summaries of the content together with links to the full version of the content and possibly other metadata.

### 4.1.3 Wikis

Wikis (WikiWikiWeb, "wiki" means "quick" in Hawaiian) are more suitable for online collaborative projects. A wiki is a type of website which allows users to easily add, remove or edit all content. Wikis were invented in 1995 by Ward Cunningham and by his definition a wiki is the simplest online database that could possibly work. Wikis are intensely collaborative and are topic related rather than user related. They represent a loosely structured set of pages, highly interlinked and also linked to other Web pages. The largest example of wiki is Wikipedia which is a free online encyclopaedia. Wikis are meant to become large shared repositories of collaboratively written knowledge. Wiki sites can be ideal for communities of practice, used for achieving collective applied learning.

Wikis have a simple syntax for authors and allow authoring via web browser and also uploading multimedia content. The main feature of wikis is collaborative editing. A page can be contributed to and edited by any user. Wikis also provide a rollback mechanism, so that the pages are versioned and the changes are transparent to all the users. Within a wiki system any concept in the text of a page can be made into an active resource very easily. Traditional wikis provide capabilities for full text search, and the recent research [Chien, Y.-H., and George, E. I. 1999] is investigating on semantic wikis which provide also semantic search and contextual navigation.

Wikis are used in different areas. They are used as encyclopaedia systems; a good example is Wikipedia which has around one million articles and is currently the largest wiki system. Another

use is for collaborative writing, in which a number geographically distributed authors can contribute on the same work simultaneously and the work can also be immediately available to readers. Wikis are also used in project and personal knowledge management, as they provide a good tool for knowledge versioning, notes and ideas repositories, knowledge base, task organization, bookmarks, etc. Other application areas for wikis are in content management systems and also for software development where collaborative tools are needed for writing documentation and for tracking software bugs. Many projects coordinate via wikis, being they public or private ones.

#### **4.1.4 Folksonomies**

A folksonomy is a collaboratively generated labelling system that enables Internet users to categorize Web content in a personal manner. In most applications, the creation of metadata has generally been approached in two different ways, either created by experts (or authors), or by normal users. The first approach is not always feasible, especially for the continuously growing data that is being produced and used. To apply the second approach, the first system was developed that allow users to tag the information with keywords. This system was called a “folksonomy” by Thomas Vander Wal in 2004 and is a combination of “folk” and “taxonomy”. The keywords, also called tags, in a folksonomy are situated in a flat space where there are no parent-child relationships between terms. The tags can be automatically clustered based on common URLs. The freely chosen tags help to improve search engine’s effectiveness because content is categorized in familiar, shared vocabularies. The vocabulary is available online to the users. With them, users can check what other users tagged and with what tags and tend to use the same vocabulary.

The best representatives for systems with folksonomies are Delicious (<http://del.icio.us>), a tool to organize web pages and Flickr (<http://www.flickr.com>), an application for photo management and web sharing. Folksonomies provide a great benefit to information retrieval, as the labels assigned by Internet users have the capacity to partially describe the subject of the Internet resources.

## **4.2 Connection to Cooper**

The COOPER system will support advanced synchronous communication mechanisms through VOI technology for establishing inter- and intra-team connections, as described in WP4. However, in addition to these services, COOPER will also use state-of-the-art communication instruments such as discussion forums, Blogs, and Wikis that will not require synchronous presence and yet will make the best possible use of state-of-the-art co-operative mechanisms.

Discussion forums and Blogs give users of COOPER a space to share their experiences and seek help from other users. Through the interactions in discussion forums or Blogs, knowledge will be created and shared. For example, when a user encounters problems in his project, he posts a question on the discussion forum. In some days, there would be some responses in the forum posted by some experienced users. They may include a number of complete solutions or links towards some external resources. These are all new knowledge, and will be incorporated in the knowledge repository of COOPER so that they could be reused in the future.

Wiki provides a more systematic way for sharing and constructing knowledge in COOPER. Whenever a project is initialized, a set of Wiki items could be constructed correspondently. The project team can then collaboratively edit the Wiki to create work plans, project deliverables and various knowledge items needed by the project. The knowledge constructed in Wiki, on the one hand, could be of high correctness, as it is agreed by the whole project team. On the other hand, it could be very complete, as it has many contributors. The knowledge items in wiki can also be input to the knowledge repository for future use.

RSS can keep the users of COOPER alerted with the updates in the discussion forum, Blog and Wiki, so that they can get timely information and take timely actions. Folksonomies allow users to collaborative annotate the knowledge items in the repository to ease the search and

browse. They all are helpful in creating a user friendly and efficient knowledge sharing environment in COOPER.

## 5 Conclusions

In this deliverable, we present the state of the art of three areas of technologies that are relevant to the knowledge sharing environment of COOPER. They include recommender systems, latent semantic analysis, and web-based knowledge sharing and co-construction systems. In order for COOPER to work as a knowledge sharing environment, we have to consider several important issues. First of all we have to establish the nature of the knowledge repository that will be available to the industrial and academic users of the platform. Such repository will initially be based upon the documents that are strictly required for supporting each project team during the various phases of their work. We will then consider the possibility of combining such internal and private resources with public resources that can be found on the Web. Relevant context information regarding each resource, such as when, where, in what purpose, by whom it was used, will also play an important role in the recommendation process.

COOPER will provide recommendations to users in the context of the projects they are involved in and also in the context of their relevant background. Users will receive suggestions regarding useful materials they should read in order to reach the projects or the team standards; also users can be recommended as experts in different fields or regarding different questions gathered from the user-platform interaction. Different recommendation techniques have to be implemented taking in consideration different user and project scenarios. We consider combining two main recommendation techniques, one based on content and the other on collaborative filtering. For the content based recommendations part we will use as main approach the latent semantic analysis.

## 6 Bibliography

Adomavicius, G., Sankaranarayanan, R., Sen, S. and Tuzhilin, A., 2005. Incorporating Contextual Information in Recommender Systems Using a Multidimensional Approach. In *ACM Transactions on Information Systems*, Vol. 23, No. 1, 103-145.

Almeida, R. B., & Almeida, V. A. F. (2004). A community-aware search engine. *WWW2004, New York, May 17-22*, 413-421. New York, USA: ACM.

Baeza-Yates, R. and Ribeiro-Neto, B. 1999. *Modern Information Retrieval*. Addison-Wesley.

Balabanovic, M. and Shoham, Y. 1997. Fab: Content-based, collaborative recommendation. *Comm. ACM* 40, 3, 66-72.

Basu, C., Hirsh, H. and Cohen, W. 1998. Recommendation as Classification: Using Social and Content-Based Information in Recommendation. In *Proceedings of the 15<sup>th</sup> National Conference in Artificial Intelligence*, Madison, WI, pp. 714-720.

Belkin, N. and Croft, B. 1992. Information Filtering and Information Retrieval. *Comm. ACM*, vol.35, no. 12, pp. 29-37.

Berry, M. W., Drmac, Z., & Jessup, E. R. (1999). Matrices, vector spaces and information retrieval. *SIAM Review*, 41(2), 335-362.

Berry, M. W., Dumais, S. T., & O'Brien, G. W. (1994, December). Using Linear Algebra for Intelligent Information Retrieval. *SIAM Review*, 37(4), 573-595. Retrieved from <http://www.cs.utk.edu/~library/TechReports/1994/ut-cs-94-270.ps.Z>

Billsus, D. and Pazzani, M. 2000. User modeling for adaptive news access. *User Modeling and User-Adapted Interaction* 10, 2-3, 147-180.

Brady M. 2005. Blogging: personal participation in public knowledge-building on the web. Chimera Working paper number: 2005, 02.

Breese, J.S., Heckerman, D., and Kadie, C. 1998. Empirical analysis of predictive algorithms for collaborative filtering. In *Proceedings of the 14<sup>th</sup> Conference in Uncertainty in Artificial Intelligence*, Madison, WI.

Burke R., *Hybrid Recommender Systems: Survey and Experiments*.

[Caron, J. \(2000\). Applying LSA to Online Customer Support: A Trial Study. Retrieved March 13, 2006 from http://www.unidata.ucar.edu/staff/caron/faqo/faqoPaper1.pdf](http://www.unidata.ucar.edu/staff/caron/faqo/faqoPaper1.pdf)

Cattell, R.B. (1966). The Scree Test for the Number of Factors. *Multivariate Behavioral Research*, 1, 245-276.

Chien, Y.-H., and George, E. I. 1999. A Bayesian model for collaborative filtering. In *Proceedings of the 7<sup>th</sup> International Workshop on Artificial Intelligence and Statistics*.

Chung-Min, C., Stoffel, N., Post, M., Bassu, D., & Behrens, C. (2001). Telcordia LSI Engine: Implementation and Scalability Issues. *Proceedings of the 11th International Workshop on Research Issues in Data Engineering (RIDE '01), Heidelberg, Germany, April 1-2*.

Claypool, M., Gokhale, A., Miranda, T., Murnikov, P., Netes, D. and Sartin, M. 1999. Combining Content-Based and Collaborative Filters in an Online Newspaper. *SIGIR'99 Workshop on Recommender Systems: Algorithms and Evaluation*. Berkley, CA.

Davis M. 2006. *Semantic Wikis for Collaboration, Information Sharing and Knowledge*

Management.

De Croock, Marcel; Pannekeet, Kees; De Vries, Fred; Sloep, Peter; Van Rosmalen, Peter. (2003) ASA: Agents for Support Activities (project plan). Heerlen: OUNL.

Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T., & Harshman, R. (1990b). Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41, 391-407.

Delgado, J. and Ishii, N. 1999. Memory-based weighting-majority prediction for recommender systems. In *ACM SIGIR'99 Workshop on Recommender Systems: Algorithms and Evaluation*.

Dessus, P., Lemaire, B., & Vernier, A. (2000). Free-Text Assessment in a Virtual Campus. *Proceedings of the CAPS'2000 conference, Paris, December 13-14*.

Dumais, S. T. (1992). *Enhancing Performance in Latent Semantic Indexing (LSI) Retrieval*. Bellcore.

Dumais, S. T. (1997, April 4). *Using Latent Semantic Indexing (LSI) for information retrieval, information filtering and other things*. Retrieved May 12, 2005, from <http://lsa.colorado.edu/papers.html>

Foltz, P. W. (1996). Latent semantic analysis for text-based research 25. *Behavior Research Methods, Instruments & Computers*, 28(2), 197-202.

Foltz, P. W., Gilliam, S., & Kendall, S. (2000). Supporting content-based feedback in online writing evaluation with LSA. *Interactive Learning Environments*, 8(2), 111-129. Retrieved from [http://www-leibniz.imag.fr/perso/s1/blemaire/public\\_html/lsa.html](http://www-leibniz.imag.fr/perso/s1/blemaire/public_html/lsa.html)

Foltz, P. W., Kintsch, W., & Landauer, T. (1998). The measurement of textual cohesion with latent semantic analysis, *Discourse Processes*, 25(2&3), 285-307.

Foltz, P. W., Laham, D., & Landauer, T. (1999). Automated Essay Scoring: Applications to Educational Technology. *Proceedings of EdMedia '99, Seattle*.

Franceschetti, D. R., Karnavat, A., Marineau, J. M. G. L., Olde, B. A., Terry, B. L., & Graesser, A. C. (2001). Development of physics test corpora for latent semantic analysis. *23th Annual Meeting of the Cognitive Science Society* 297-300.

Freeman, J. T., Thompson, B. T., & Cohen, M. S. (2000). Modeling and Diagnosing Domain Knowledge Using Latent Semantic Indexing. *Interactive Learning Environments*, 8(3), 187-209.

Getoor, L. and Sahami, M. 1999. Using probabilistic relational models for collaborative filtering. In *Workshop on Web Usage Analysis and User Profiling (WEBKDD'99)*.

Ghita S., Nejd W. and Paiu R. Semantically Rich Recommendations in Social Networks for Sharing and Exchanging Semantic Context, Proceedings of the 2nd European Semantic Web Conference Workshop on Ontologies in P2P Communities, ESWC, Greece, May 2005

Giles, J. T., Wo, L., & Berry, M. W. (2001). GTP (General Text Parser) Software for Text Mining. In *Statistical Data Mining and Knowledge Discovery* (chap. 27). CRC Press. Retrieved from <http://www.cs.utk.edu/~berry/papers02/GTPchap.pdf>

Godwin-Jones R., 2003. Emerging technologies. Blogs and Wikis: Environments for On-line Collaboration. *"Language Learning & Technology"* May 2003, Vol. 7, No. 2, 12-16.

Goldberg, K., Roeder, T., Gupta, D. and Perkins, C. 2001. Eigentaste: A constant time collaborative filtering algorithm. *Information Retrieval Journal*, 4, 2, 133-251.

Graesser, A. C., Hu, X., Olde, B. A., Ventura, M., Olney, A., Louwerse, M., et al. (2005). Implementing Latent Semantic Analysis in Learning Environments with Conversational Agents

and Tutorial Dialog. In W. D. & S. Gray (Ed.), *24th Annual Meeting of the Cognitive Science Society* 37. Mahwah, NJ: Erlbaum.

Hill, W., Stead, L., Rosenstein, M. and Furnas, G. 1995, Recommending and evaluating choices in a virtual community of use. In *CHI'95: Conference Proceedings on Human Factors in Computing Systems*, Denver, CO, pp. 194-201.

Hofmann, T. (1999). Probabilistic Latent Semantic Analysis. *Uncertainty in Artificial Intelligence*. Retrieved March 9, 2006,

Kanejiya, D., Kumar, A., & Prasad, S. (2003). Automatic Evaluation of Students' Answers using Syntactically Enhanced LSA. *Human Language Technology Conference (HLT-NAACL 2003) Workshop on Building Educational Applications using NLP*. Edmonton, Canada.

Kester, L., Sloep, P., Brouns, F., Van Rosmalen, P. De Vries, F. De Croock, M. Koper, R. (submitted). Enhancing Social Interaction and Spreading Tutor Responsibilities in Bottom-Up Organized Learning Networks.

Kintsch, E., Steinhart, D., Stahl, G., LSA research group, Matthews, C., & Lamb, R. (2000). Developing summarization skills through the use of LSA-based feedback 30. *Interactive Learning Environments*, 8(2), 87-109.

Kintsch, W. (1998). The representation of knowledge in minds and machines. *International Journal of Psychology*, 33(6), 411-420.

Kintsch, W. (2001). Predication. *Cognitive Science*, 25, 173-202.

Kintsch, W. (2002a). On the notions of theme and topic in psychological process models of text comprehension. In M. Louwerse & W. van Peer (Eds.), *Thematics: Interdisciplinary studies* (pp. 157-170). Amsterdam, Netherlands: John Benjamins Publishing Company.

Kintsch, W. (2002b). The Potential of Latent Semantic Analysis for Machine Grading of Clinical Case Summaries. *Journal of Biomedical Informatics*, 35, 3-7.

Kintsch, W., & Bowles, A. (2002). Metaphor comprehension: What makes a metaphor difficult to understand? *Metaphor and Symbol*, 4(17), 249-262. Retrieved from <http://lsa.colorado.edu/papers.html>

Konstan, J. A., Miller, B. N., Maltz, D., Herlocker, J. L., Gordon, L. R., and Riedl, J. 1997. GroupLens: Applying collaborative filtering to Usenet news. *Comm. ACM* 40, 3, 77-87.

Koper, R., van Bruggen, J., Rusman, E., & Giesbers, B. (2005). *Learning Technology Development Programme - Positioning in learning networks*.

Laham, D. (1997). Latent Semantic Analysis approaches to categorization. *Proceedings of the 19th annual meeting of the Cognitive Science Society*, 979.

Laham, D., Bennett, W., & Derr, M. (2002). *Latent Semantic Analysis for Career Field Analysis and Information Operations*. Retrieved January 24, 2006, from <http://www.k-a-t.com/papers/ab-careerField2002.shtml>

Laham, D., Bennett, W., & Landauer, T. K. (2000). An LSA-Based Software Tool for Matching Jobs, People, and Instruction. *Interactive Learning Environments*, 8, 171-185.

Landauer, T. K. (2002b). On the computational basis of learning and cognition: Arguments from LSA. In B. H. Ross (Ed.), *The psychology of learning and motivation: Advances in research and theory*, Vol. 41 (rep. No. 2002-10361-005, pp. 43-84). San Diego, CA, US: Academic Press.

Landauer, T. K. (9-8-2002a). Applications of Latent Semantic Analysis. *24th Annual Meeting of the Cognitive Science Society*.

Landauer, T. K., & Dumais, S. T. (1997). A solution to Plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological Review*, 104(2), 211-240.

Landauer, T. K., Foltz, P. W., & Laham, D. (1998). An introduction to latent semantic analysis. *Discourse Processes*, 25(2-3), 259-284.

Landauer, T. K., Laham, D., & Foltz, P. W. Jordan, M. I., Kearns, M. J., & Solla, S. A. (Eds.). (1998). Learning human-like knowledge by Singular Value Decomposition: A progress report. *Advances in Neural Information Processing Systems*, 10, 45-51. Retrieved from <http://lsa.colorado.edu/papers.html>

Lemaire, B., & Dessus, P. (2001). A System to Assess the Semantic Content of Student Essays. *Journal of Educational Computing Research*, 24(3), 305-320.

Letsche, T. A. (1997). Large-Scale Information Retrieval with Latent Semantic Indexing. *Informatics and Computer Science*, 100, 105-137. Retrieved from <http://www.cs.utk.edu/~berry/lis++/>

Magliano, J. P., Wiemer-Hastings, K., Millis, K. K., Munoz, B. D., & McNamara, D. (2002). Using latent semantic analysis to assess reader strategies. *Behavior Research Methods, Instruments & Computers*, 34(2), 181-188.

Miller, T. (2003). Essay assessment with latent semantic analysis. *Journal of Educational Computing Research*, 29(4), 495-512.

Mooney, R.J., Bennett, P.N. and Roy, L. 1998. Book recommending using text categorization with extracted information. In *Recommender Systems. Papers from 1998 Workshop. Tech. Rep. WS-98-08*. AAAI Press

Nakamura, A. and Abe, N. 1998. Collaborative filtering using weighted majority prediction algorithms. In *Proceedings of the 15<sup>th</sup> International Conference on Machine Learning*.

Nakov, P., Valchanova, E., & Angelova, G. (2003). Towards deeper understanding of the LSA Performance. *Recent Advances in Natural Language processing (RANLP'2003)*311-318.

Norris, Donald M., Mason, J., Robson, R., Lefrere, P., and Collier, G. A Revolution in Knowledge Sharing. *EDUCAUSE REVIEW*, vol. 38, no. 5 (September/October 2003).

Olde, B. A., Franceschetti, D. R., Karnavat, A., Graeser A.C., & Tutoring Research Group. (2002). The Right Stuff: Do You Need to Sanitize Your Corpus When Using Latent Semantic Analysis. *24th Annual Meeting of the Cognitive Science Society*708-713. Fairfax.

Panzani, M.J. 1999. A framework for Collaborative, Content-Based and Demographic Filtering. *Artificial Intelligence Review*, 13 (5/6), 393-408.

Pazzani, M. and Billus, D. 1997. Learning and revising user profiles: The identification of interesting web sites. *Machine Learning*, 27, 313-331.

Pennock, D. M. and Horovitz, E. 1999. Collaborative filtering by personality diagnosis: A hybrid memory- and model-based approach. In *IJCCAI'99 Workshop: Machine Learning for Information Filtering*.

Quesada, J. F. (2003). Introduction to Latent Semantic Analysis and Latent Problem Solving Analysis. In *Latent Problem Solving Analysis (LPSA): A computational theory of representation in complex, dynamic problem solving tasks* (pp. 22-35) (chap. 2). Retrieved from <http://www.andrew.cmu.edu/user/jquesada/dissertation/>

Quesada, J. F., Kintsch, W., & Gomez, E. (2001). A Computational Theory of Complex Problem Solving Using the Vector Space Model (part I): Latent Semantic Analysis, Through the

Path of Thousands of Ants. *Cognitive research with Microworlds*, 43(84), 117-131. Retrieved from <http://lsa.colorado.edu/papers.html>

Rehder, B., Schreiner, M. E., Wolfe, M. B., Laham, D., Landauer, T. K., & Kintsch, W. (1998). Using latent semantic analysis to assess knowledge: some technical considerations *Discourse Processes*, 25, 337-354.

[Rosario, B. \(2000\). Latent Semantic Indexing: An Overview \(INFOSYS 240\). Retrieved October 10, 2005, from http://www.sims.berkeley.edu/~rosario/projects/LSI.pdf](http://www.sims.berkeley.edu/~rosario/projects/LSI.pdf)

Stahl, G. Allowing learners to be articulate: incorporating automated text evaluation into collaborative software environments. *Proposal to the McDonnell foundation*, 1997.

Turney, P. D., Litmann, M. L., Bigham, J., & Shnayder, V. (2003). Combining Independent Modules to Solve Multiple-Choice Synonym and Analogy Problems. In G. Angelova, K. Bontcheva, R. Mitkov & Nicolov, N. (Eds.), *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP-03), Borovets, Bulgaria, september 10-12, 2003*, 482-389.

Van Bruggen, J. M., Rusman, E., Giesbers, B. & Koper, R. (submitted). Latent semantic analysis of small-scale corpora for positioning in learning networks.

Van Bruggen, J., Sloep, P., van Rosmalen, P., Brouns, F., Vogten, H., Koper, R., & Tattersall, C. (2004). Latent semantic analysis as a tool for learner positioning in learning networks for lifelong learning. *British Journal of Educational Technology*, 35(6), 729-738.

Wade-Stein, D., & Kintsch, E. (2003). *Summary Street: Interactive Computer Support for Writing* (03-01(2003)). Colorado: University of Colorado. (Sectie W). Retrieved May 12, 2005, from [http://www-leibniz.imag.fr/perso/s1/blemaire/public\\_html/lsa.html](http://www-leibniz.imag.fr/perso/s1/blemaire/public_html/lsa.html)

Wiemer-Hastings, P. (1999). How latent is Latent Semantic Analysis? 29. Proceedings of the Sixteenth International Joint Congress on Artificial Intelligence, 932-937. San Francisco: Morgan Kaufmann.

Wiemer-Hastings, P., & Graesser, A. C. (2000). Select-a-Kibitzer: a computer tool that gives meaningful feedback on student compositions, *Interactive Learning Environments*, 8(2), 149-169.

Wiemer-Hastings, P., & Zipitria, I. (2001). Rules for syntax, vectors for semantics. *Proceedings of the Twenty-third Annual Conference of the Cognitive Science Society*, 1112-1117. Mahwah, NJ: Lawrence Erlbaum Associates.

Wiemer-Hastings, P., Wiemer-Hastings, K., & Graesser, A. C. (1999). Improving an intelligent tutor's comprehension of students with Latent Semantic Analysis. In S. P. Lajoie & Vivet M. (Eds.), *Artificial Intelligence in Education*. Amsterdam: IOS Press. 535-542.

Wild, F., Stahl, C., Stermsek, G., & Neumann, G. (2005). Parameters driving effectiveness of automated essay scoring with LSA. *Proc. of the 9th International Computer Assisted Assessment Conference (CAA), Loughborough, July, 2005*, 485-494.

Wild, F., Stahl, C., Stermsek, G., Peña, Y., & Neumann, G. (2005). Factors influencing effectiveness in automated essay scoring with LSA. *Proc. of the 12th International Conference on Artificial Intelligence in Education (AIED), Amsterdam, July, 2005* IOS Press. 485-494.

Wolfe, M. B. W., & Goldman, S. R. (2003). Use of latent semantic analysis for predicting psychological phenomena: Two issues and proposed solutions. *Behavior Research Methods, Instruments & Computers*, 35(1), 22-31.

Wolfe, M. B. W., Schreiner, M. E., Rehder, B., Laham, D., Foltz, P. W., Kintsch, W., et al. (1998). Learning from text: Matching readers and texts by latent semantic analysis. *Discourse*

*Processes*, 25(2-3), 309-336.

Yu, C., Cuadrado, J., Ceglowski, M., & Payne, J. S. (4-2-2005). *Patterns in Unstructured Data - Discovery, Aggregation, and Visualization*. Retrieved May 12, 2005, from [http://research.nitle.org/lsi/cover\\_page.htm](http://research.nitle.org/lsi/cover_page.htm)

Zampa, V., & Lemaire, B. (2002, October 15). Latent Semantic Analysis for User Modelling. *Journal of Intelligent Information Systems*, 18(1), 15-30. Retrieved from [http://www.leibniz.imag.fr/perso/s1/blemaire/public\\_html/lisa.html](http://www.leibniz.imag.fr/perso/s1/blemaire/public_html/lisa.html)